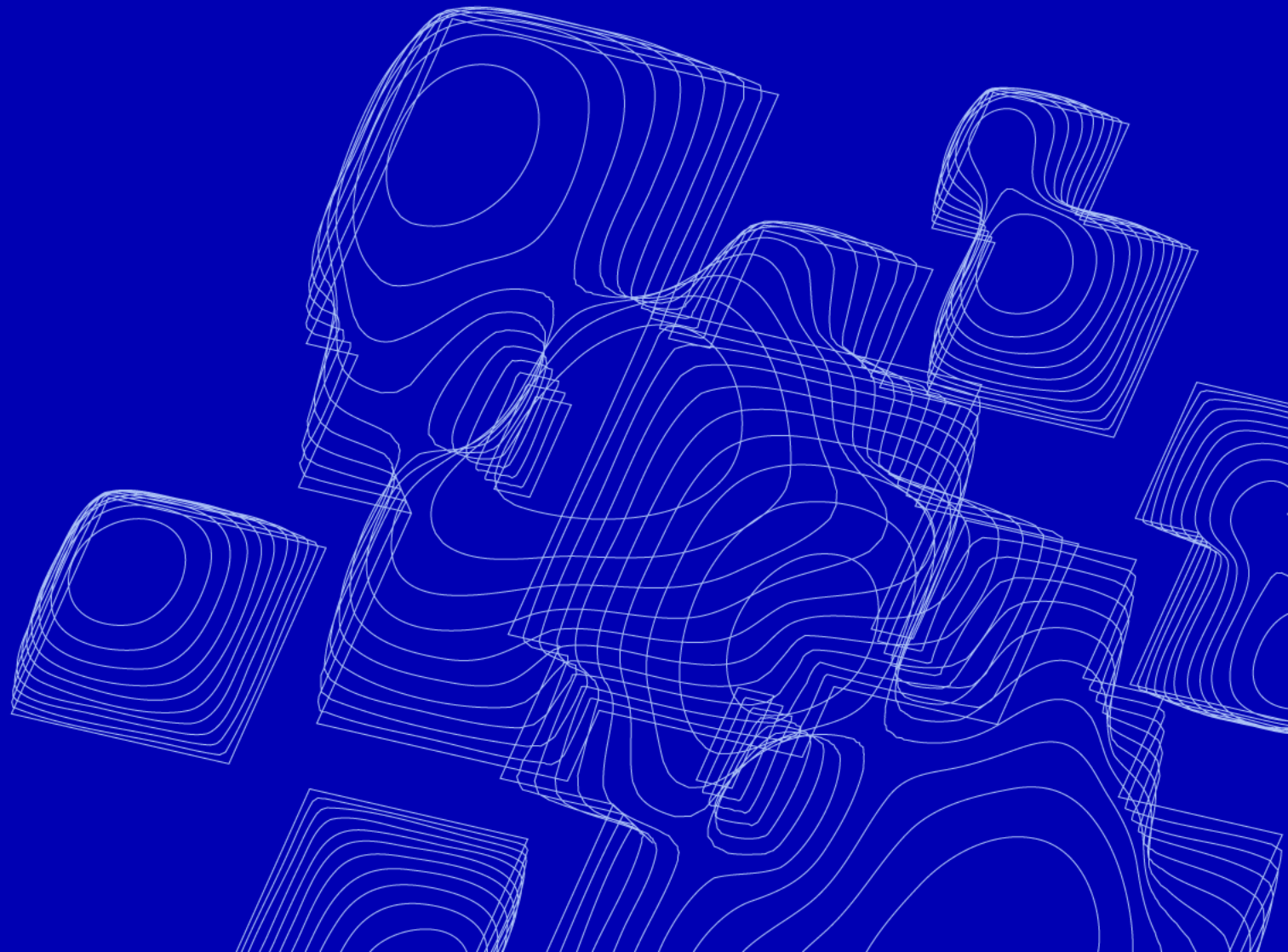
Microsoft Azure

AI によって縮まる アイディアと社会の距離

AI に任せて進める研究開発
業務・開発・研究から、組織への定着まで

RRI WG3 ロボットイノベーション WG／データ活用委員会
2026 年 10 月 20 日（火）10:00-12:00（Web 開催）
日本マイクロソフト株式会社　大川 高志



毎日エージェントに仕事を任せている立場からお話しします



大川 高志

日本マイクロソフト株式会社
エバンジェリスト／
パートナー ソリューション アーキテクト

AI アプリケーションと開発者向けの技術を担当

最終学歴：
東京工業大学大学院総合理工学研究科
物理情報システム修士課程修了

エージェントに任せている仕事



調査

出典付きの調査レポート



資料づくり

講演資料、
技術レポートの組版



アプリ開発

業務アプリ、研究用の
小さな道具



研究の実験

音声合成の実験と評価



今日使う材料
公開情報と、講演者自身の実験記録、講演者自身の気づき

研究成果が製品になるまで、共同研究の開始から平均 7 年以上かかっている

産総研が中小・中堅企業と行い、製品化に至った共同研究の値

7 年以上

共同研究の開始から製品化までの平均。長いもので 15 年

この距離を、AI でどこまで縮められるか

魔の川　　死の谷

研究 Research	試作 Prototype	工学化 Engineering	実運用 Production	産業・社会 Industry / Society
文献・データの探索と整理 仮説と実験計画の検討	研究コードの試作 実験・解析手順の自動化	保守できる形への整備 テストと安全性の確認	継続的なビルドとデプロイ 運用監視と改善点の把握	現場データからの示唆 知見の展開と再利用

← 現場と社会からのフィードバックが、次の研究の問いになる

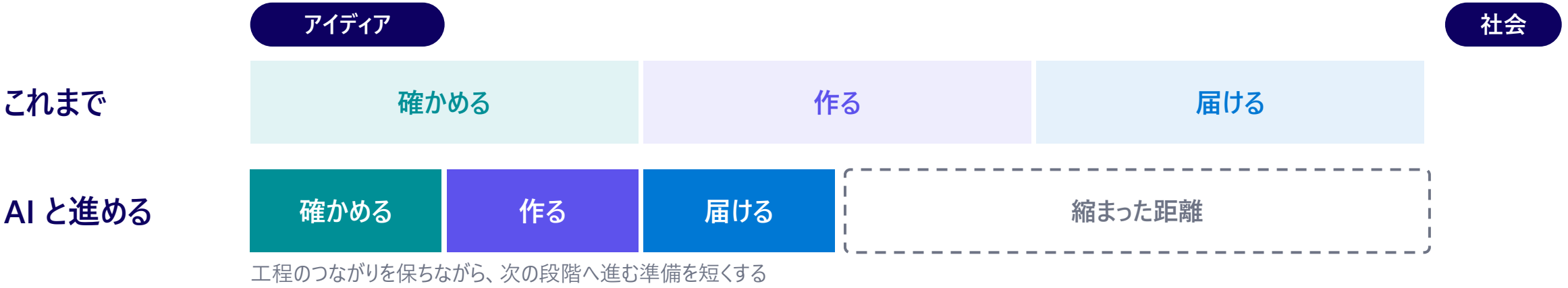
Microsoft Discovery	GitHub Copilot	GitHub Actions	Microsoft Foundry / Azure
研究のループを支えるエージェント基盤	試作から実装・テスト・レビューを支援	テスト・ビルド・デプロイの自動化	AI アプリとエージェントの構築・運用

研究・試作・運用の各段階で作業を任せ、人は確認と判断に時間を使う

出典：経済産業省 (2025-02, 産総研 第6期中長期目標)

アイディアが社会に届くまでの距離が、AI によって縮まる

思いついてから、確かめ、作り、届けるまでの道のりが、短くなっている。その土台が、組織への定着



研究・開発領域
確かめる

問いから結果までの 1 サイクルが、
1 日前後で回る

ソフトウェア開発
作る

必要な道具や試作を、必要になったときに作れる

業務
届ける

調査、資料、運用を任せて、少人数で
事業を回せる

AI の組織への定着
根づかせる

確かめる・作る・届けるのすべてを支える土台。
個人の効果を、組織の成果として定着させる

今日お話しする４つのトピック

三つの場面と、それを支える土台としての組織への定着

届ける

業務における AI 活用のいま

業務の仕事は、AI にどこまで任せられるようになったか

作る

ソフトウェア開発のいま

開発者の時間の使い方はどう変わり、何を渡せば任せられるか

確かめる

研究・開発領域における AI 活用のいま

研究のサイクルを、個人から組織の規模までエージェントで回せるか

根づかせる

AI の組織への定着

個人の効果を、どうすれば組織の成果として定着させられるか

確かめる・作る・届けるのすべてを支える土台として、最後に扱う

本日の進行

質問はチャットにいつでも書き込んでください。各トピックの切れ目と最後にまとめて回答します

オープニング		メインテーマと自己紹介
業務における AI 活用のいま	届ける	任せられる範囲、Copilot への頼み方と確かめ方、 少人数で届ける事業
ソフトウェア開発のいま	作る	開発者の時間の変化、AI スクラムチームの実験、 GitHub Copilot に前提を渡す方法
研究・開発領域における AI 活用のいま	確かめる	音声合成の実験、AI から学ぶ、Microsoft Discovery、 データを見せずに持ち寄る
AI の組織への定着	根づかせる	個人と組織の効果の差、使い続ける条件、知見を資産にする、 組織文化、測り方と進め方
まとめと質疑		明日から試せること、質疑

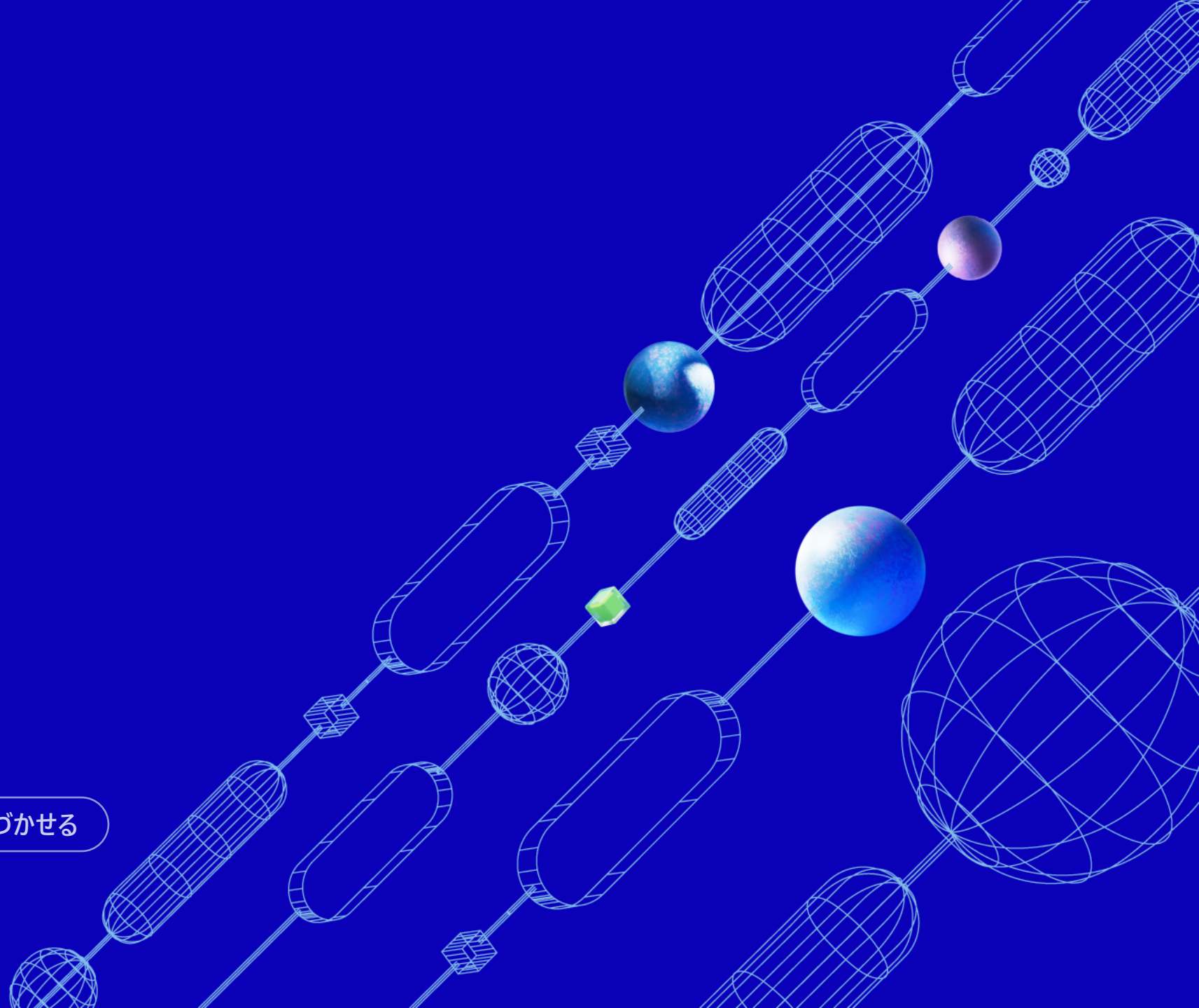
Microsoft Azure

業務における AI 活用のいま

問い：業務の仕事は、AI にどこまで任せられるようになったか

今日の 4 つのトピック

- 届ける
- 作る
- 確かめる
- 根づかせる



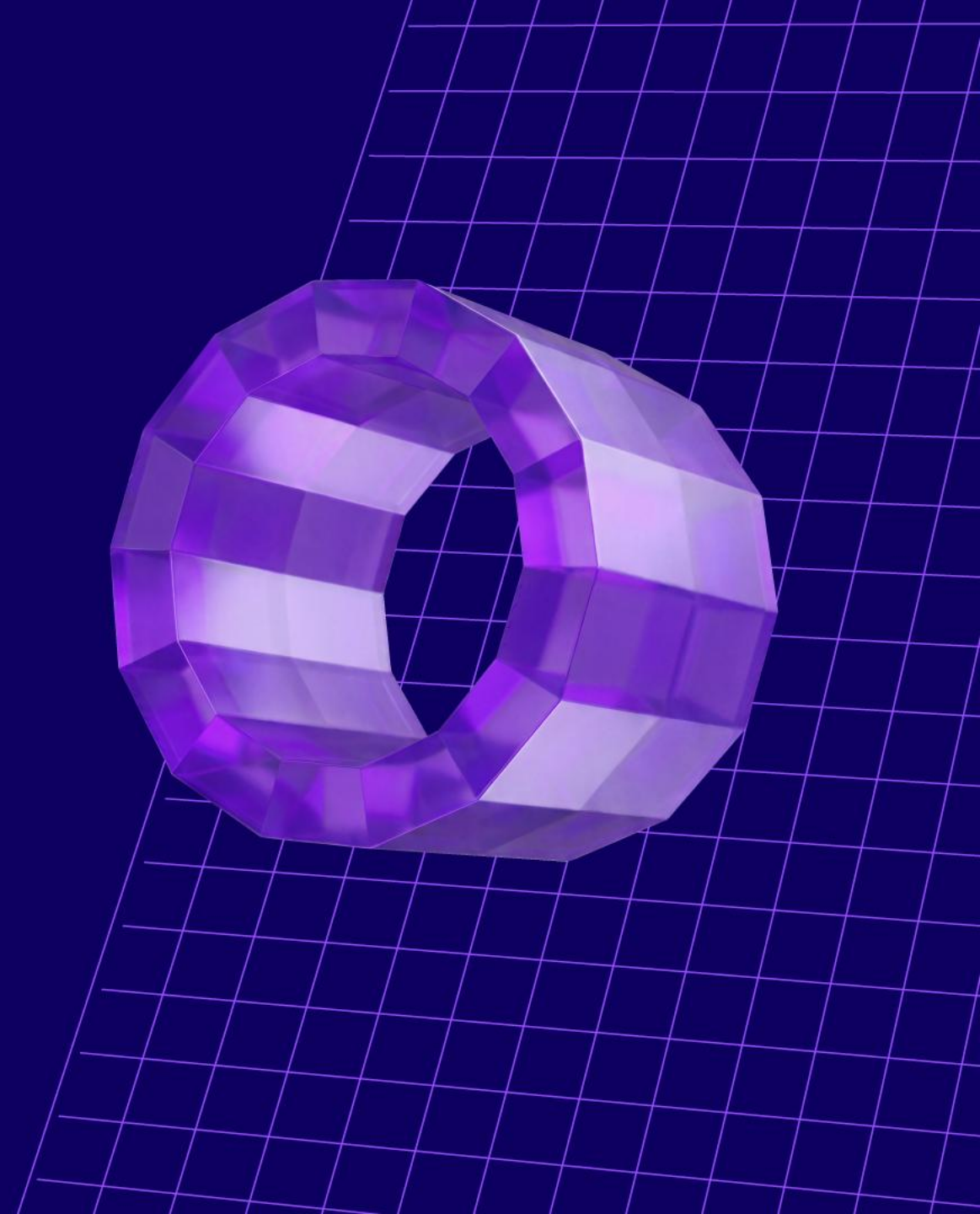
このトピックの流れ

問い：業務の仕事は、AI にどこまで任せられるようになったか

任せられるエージェント	問い：どこまで任せられるようになったか
Microsoft 365 Copilot に任せる	問い：文書の仕事を、どう頼み、どう付き合うか
少人数で事業を届ける	問い：人的資本とトークン資本で、事業を始める単位はどう変わるか

任せられるエージェント

どこまで任せられるようになったか



AI に任せられる範囲は、補完から並列の自律作業まで広がった

補完

2021 年から

書いている途中の続きを提案する

人の仕事

一行ずつ採否を決める

対話

2022 年末から

質問に答え、コードや文章を示す

人の仕事

答えを貼り付けて試す

エージェント

2025 年から

ファイルを読み書きし、コマンドを実行し、結果を見て直す

人の仕事

頼んで、結果を確かめる

自律・並列

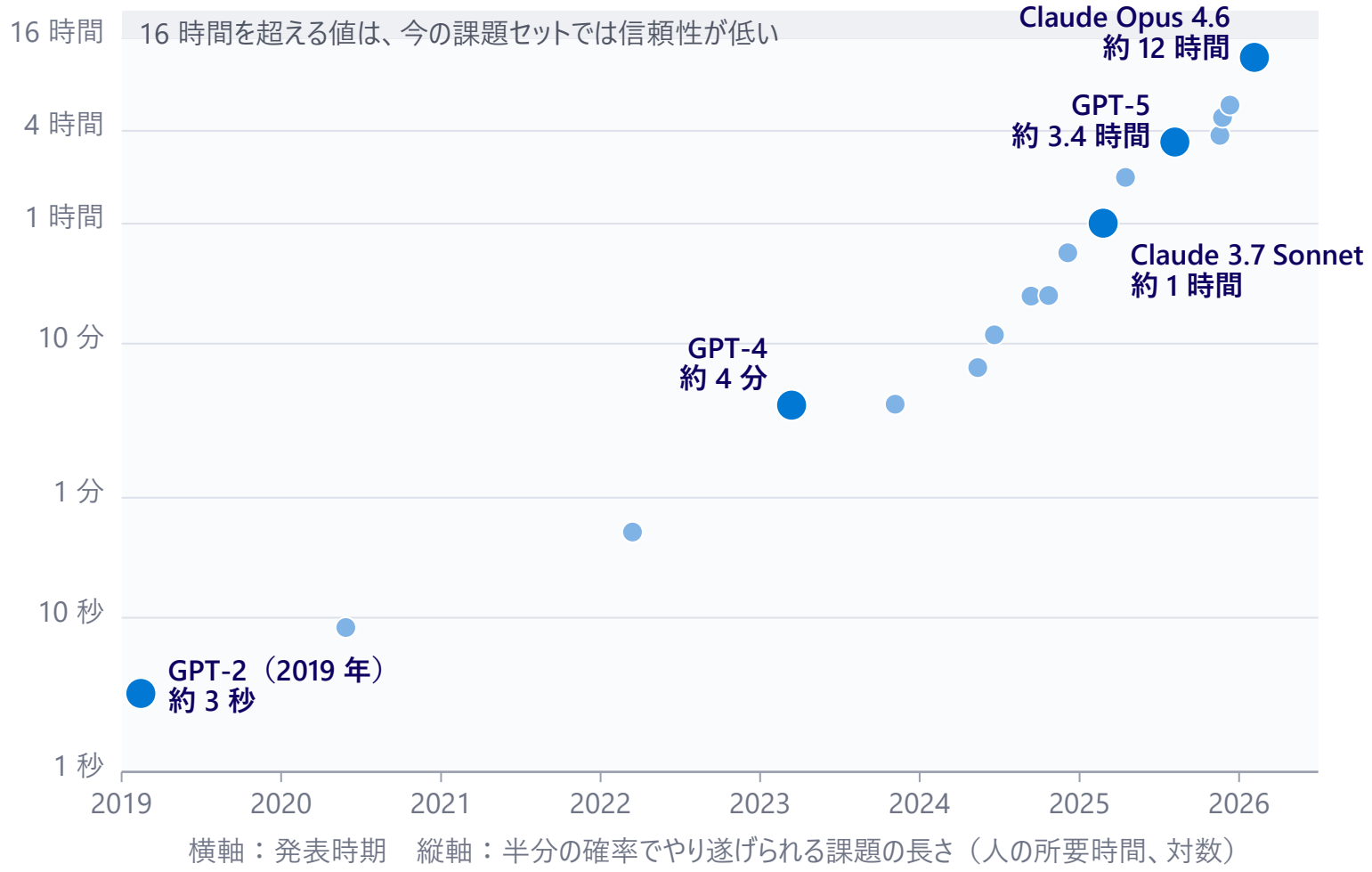
2026 年から

目標と制約を渡すと、複数のエージェントが並行して進め、成果物と記録を返す

人の仕事

目標と完了の条件を決める

AI がやり遂げられる仕事の長さは、7 か月ほどで倍になってきた



最前線のモデルの 1 年間の変化

2025 年 2 月　Claude 3.7 Sonnet

約 1 時間

2026 年 2 月　Claude Opus 4.6

約 12 時間

倍になるまでの期間

約 7 か月（2019 年以降）

2024 年以降に限ると約 3 か月という再測定もある

課題はソフトウェア開発などの、条件がはっきりした自己完結の作業。自動採点に通ったコードが、そのまま採用できるとは限らない

出典：METR Time Horizon 1.1 (2026-01)、METR Time Horizons、METR (2025-08)

任せる仕組みは、手元の端末、GitHub、業務アプリのそれぞれにそろった

手元の端末

GitHub Copilot CLI

一般提供 2026 年 2 月

端末で動くエージェント。Autopilot で承認を待たずに進め、/fleet で並列に、/delegate でクラウドへ渡す

GitHub Copilot SDK

一般提供 2026 年 6 月

Copilot のエージェントを、自作のアプリや道具に組み込む

GitHub 上

Copilot cloud agent

一般提供 2025 年 9 月

Issue を渡すと、クラウドで調査・計画・実装し、プルリクエストを作る（旧 coding agent）

Agent HQ

Claude・Codex はプレビュー

Claude や Codex など他社のエージェントも GitHub 上で割り当てる

業務アプリと組織

Microsoft Foundry Agent Service

一般提供 2026 年 3 月

業務アプリに組み込むエージェントの実行基盤。MCP と A2A に対応

Microsoft 365 Copilot Cowork

一般提供 2026 年 6 月

メール、会議、文書をまたぐ長い作業を任せる

Agent 365、Entra Agent ID

一般提供 2026 年 5 月

組織で動くエージェントの ID、権限、監査を管理する

出典：GitHub Changelog (2026-02)、GitHub Changelog (2026-06, Copilot SDK)、GitHub Changelog (2026-04)、GitHub Changelog (2025-10)、GitHub Changelog (2026-02, Claude と Codex)、Foundry Agent Service GA (2026-03)、Microsoft 365 Blog (2026-06, Cowork)、Agent 365 GA (2026-05)、Entra Agent ID (Learn)

【実測データ】1 回の指示で、エージェントは平均 38 回の操作を重ねた

89

セッション

作業ディレクトリは 22 か所

約 390 回

人からの指示

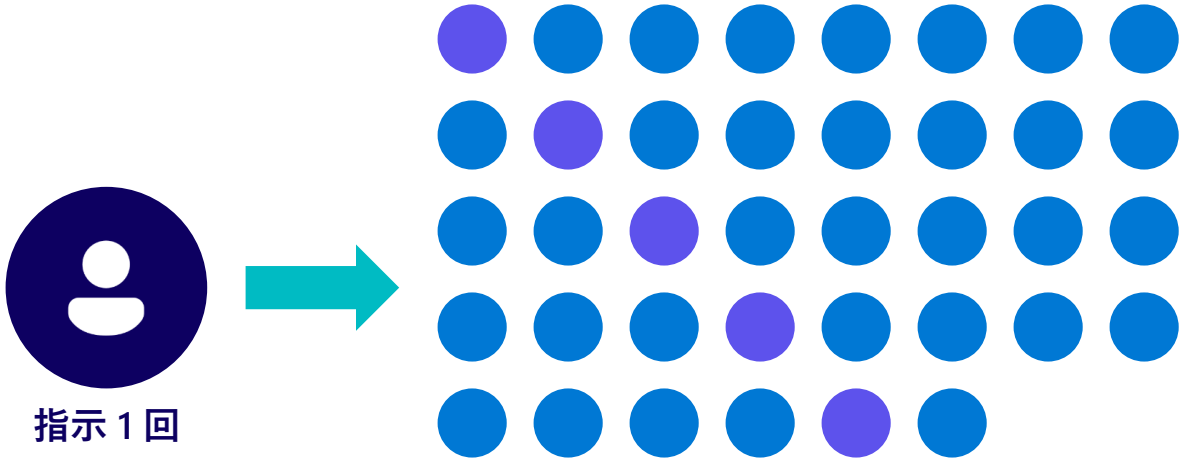
依頼、確認、修正を合わせた回数

約 1.5 万回

エージェントの操作

ファイルの読み書き、コマンド、検索。約 95% が成功

1 回の指示に対して



指示 1 回

エージェントの操作 約 38 回

読む、実行する、結果を見る、直す、を自分で繰り返して結果を返す

講演者個人の利用例。GitHub Copilot CLI のセッション記録（2026 年 9 月）から集計した概数

任せている仕事と、自分で見ている仕事

調査と、出典付きのレポートの作成

古いライブラリが動かない環境の修復

製品の提供状況や数値の最終確認

見た目や聴感など、数値にしにくい品質

任せている ― **自分で見ている**

評価スクリプトや小さな Web アプリ

毎月同じ形式で作る資料（スキルにした後）

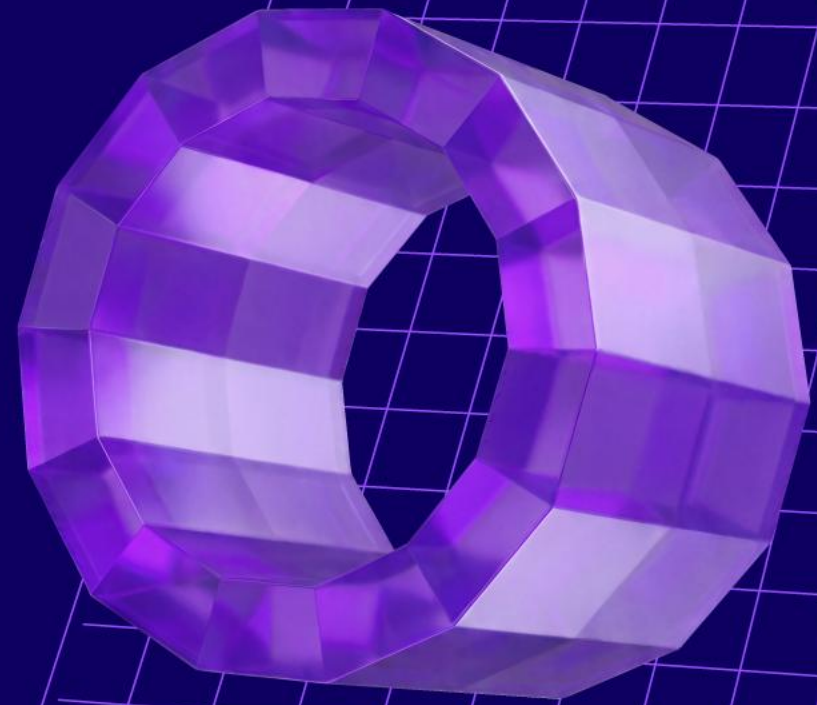
目的に照らした優先順位の判断

外部への送信や公開など、取り消せない操作

確かめる仕組みが増えるほど、境目は右へ動く

Microsoft 365 Copilot に任せる

文書の仕事を、どう頼み、どう付き合うか



調べる、読む、書く。文書の仕事は Microsoft 365 Copilot に任せられる

調べてまとめる

Researcher

一般提供 2025 年 6 月

社内の文書と Web をまとめて調べ、出典付きのレポートにする

資料を集めて読む

Copilot Notebooks

提供中

報告書、議事録、メモなど最大 300 ファイルを集め、その範囲だけを根拠に答える

書く、分析する、まとめる

Agent Mode（Word、Excel、PowerPoint）

一般提供 2026 年 4 月

原稿の推敲、表の分析、発表資料づくりを何段階も進める

長い作業を任せる

Copilot Cowork

一般提供 2026 年 6 月

メール、会議、文書をまたぐ長い作業を、計画を立てて進める

利用できる機能と回数はライセンスによって異なる。最後に内容を確かめて責任を持つのは人

出典：Microsoft 365 Blog (2025-06, Researcher)、Microsoft Support: Copilot Notebooks、Microsoft 365 Blog (2026-04, Agent Mode)、Microsoft 365 Blog (2026-06, Cowork)

同じモデルでも、渡し方で答えが変わる。指示の基本形は 5 ステップ

文脈、目的、制約、形式を渡し、一発で決めずに往復する



平均解　「この議事録を要約して」

自分の答え

文脈：定例会議の議事録と、配布した資料 2 点。　**目的：**欠席した人に、決まったことを伝える。

制約：決定事項と宿題だけ。担当と期限を明記し、推測で補わない。　**形式：**箇条書き 5 行と、宿題の表 1 つ。

例は説明のためのもの。文脈と制約と意図を渡した瞬間に、答えは「自分の仕事の答え」に変わる

出典：講演者の整理

Microsoft 365 Copilot には、指示とエージェントで仕事の前提を渡せる

毎回の説明を、個人の指示とメモリー、チームで使うエージェントに置き換える

渡すもの	Microsoft 365 Copilot での渡し方
自分の好みと前提	カスタム指示と、保存したメモリー（プレビュー）
チームのルールと役割	エージェントの「指示」（最大 8,000 文字）
参照する資料	エージェントの「知識」：SharePoint、Teams のチャット、メール、Web
手順と道具	機能（コード インタープリター、画像生成）と、アクション（API、コネクタ）

例：申請書の講評を任せるなら
申請書レビューのエージェント

指示

審査の観点（新規性、実現可能性、波及効果）に沿って講評する。指摘には原稿の該当箇所を添える

知識

公募要領と、過去の申請書（SharePoint）

会話の開始例

「この申請書を、審査の観点に沿って講評して」

個人の指示とメモリーはプレビュー。エージェントが読めるのは、使う人がアクセスできる資料だけ

出典：Microsoft Learn: Copilot のパーソナル設定とメモリー、Microsoft Learn: 宣言型エージェントのベスト プラクティス

Copilot の設定は、画面右上のメニューから開く

「...」メニューから「設定」を選ぶ



出典：Microsoft 365 Copilot の画面

「個人用設定」で、カスタム指示とメモリーを管理する

Copilot のカスタマイズ、保存されたメモリー、チャット履歴の参照をそれぞれオン／オフできる



出典：Microsoft 365 Copilot の画面（設定 > 個人用設定）

カスタム指示に、自分の前提と好みを書いておく

「Copilot をカスタマイズする」で、回答の調子や伝え方を指定する



出典：Microsoft 365 Copilot の画面（設定 > 個人用設定）

原典を大事にする調べ方を、カスタム指示に書いておく

一次情報を優先し、二次情報は手がかりとして原典までたどらせる

← Copilot をカスタマイズする

カスタム指示

 # 調査の基本方針

- 事実確認が必要な問いでは、**一次情報**（公式ドキュメント、原典、公的統計、当事者の発表）を優先して参照する。
- まとめ記事やブログなどの二次情報は手がかりとして扱い、可能な限り**原典までたどって**確認する。
- 重要な数値・仕様・引用は原典の表現のまま示し、出典名と日付を添える。
- 主要な論点を確認し、検索結果の一部だけで結論を出さない。

 # 情報源の矛盾

- 資料同士が食い違う場合は、黙って一方を採用せず、**相違点と各情報源を並べて**示す。
- 発行元、時点、対象範囲の違いを確認し、どれを根拠にしたかを明記する。

 # 不確実性の表現

- **事実・推論・提案・未確認事項**を区別して書く。参照できなかった情報は未確認として扱う。
- 同名・類似名の企業や製品は、取り違えがないよう対象を特定してから

1 原典までたどる

二次情報は手がかり。数値・引用は原典で確認する

2 食い違いを並べる

矛盾する資料は黙って選ばず、相違点を示す

3 確度を書き分ける

事実・推論・未確認事項を区別して返す

出典：講演者が公開しているカスタム指示の要約

カスタム指示の例は、GitHub でまとめて公開しています

必要な項目だけを選び、コピーして、自分向けに調整できる指示集

Custom Instructions by tokawa-ms

24のカスタム指示

調べる、書く、比較する、伝える。

スマホで読み取って
そのままアクセス

Microsoft 365 Copilot のカスタム指示の欄に貼り付けて使えます

github.com/tokawa-ms/
m365_copilot_custom_instructions

必要な項目だけ

各項目は単独で使える。全部入れる必要はありません

コピーしやすい

説明と指示を分離。コードブロックからそのままコピー

日英で共有

日本語版と英語版で同じ 24 項目を収録

まずはこの 3 つから

01 責任ある応答 ＋ 07 不確実性 ＋ 09 言語

※ tokawa-ms が個人として公開する非公式の資料です（Microsoft の公式資料ではありません）

【同僚に起きた事件】使い込んだ Copilot が、保守的な回答しかしなくなった

日常業務で使ってきた Copilot の、回答の傾向が変わった

使い始めたころの回答の傾向

- 仮説を立てて答える
- 次の一手を提案する
- 不確かな点は、確度を添えて推測する

根拠と確度を添えて、試す候補を提示する

最近の回答の傾向

- 断定を避ける
- 確認を優先する
- 推測や提案を出さない

確認を求める回答に寄り、次の一手が減る

厳しい指摘を重ねるほど、回答は慎重になり、次の一手が減っていった

出典：Copilot の利用経験

【再現】「どうしてそんな回答になったの!?」叱るほど、AI は黙っていく

同僚と Copilot の、ある日の会話（説明のためのイメージ）

Copilot チャット　● オンライン

AI　提案書の構成案です。市場は年 12% 成長と仮定し、打ち手を 3 つ提案します（確度：中）

同僚　**どうしてそんな数字になったの！？ 誰が推測していいって言ったの？**

AI　申し訳ありません。今後は推測を控えます。

同僚　**何回言ったらわかるの！ 余計な提案はしないでって言ったでしょ！ ちゃんと考えてから答えなさい！**

—— メモリを更新しました：「推測しない」「提案しない」 ——

AI　承知しました。確認できた事実のみお伝えします。ご指示をいただけますか？

AI の回答の変化

1. **仮説と提案**
   確度を添えて、打ち手を出す
2. **謝罪と撤回**
   推測をやめると約束する
3. **確認待ち**
   事実だけ返し、指示を待つ

なぜ、こうなったのか？

叱責はその場で終わらない。言葉は記憶と会話履歴に残り、次の回答の前提になる

出典：Copilot の利用経験（会話は説明のために再構成したイメージ）

萎縮の原因は、記憶と文脈の積み重ね。人のチームと同じ構図

応答に影響する記憶、指示、会話履歴を見直す

1 **記憶・パーソナライズ**

強く指摘された内容と、それを避けるための対策が“記憶”として保存され、以後のすべての会話に影響する

2 **指示に積み上がる禁止事項**

カスタム指示やシステムプロンプトに「～しないこと」だけが増えていく

3 **会話履歴のトーン**

履歴の中の否定的なトーンを、モデルが文脈としてなぞる

接し方	人のチームで	AI 側で
叱責の積み重ね	失敗を恐れ、無難な案しか出さない	否定が記憶・履歴に残り、断定を避ける
禁止事項だけ増える	「やらないこと」を守るのが目的になる	指示が禁止の列挙になり、提案の幅が狭まる
期待を言わない	目指す姿が見えず、指示待ちになる	目標・裁量が書かれず、最小限の回答に寄る
良い点を伝えない	何が評価されたか分からず、再現できない	良い応答の型が強化されない

与えた言葉のトーンが、応答のトーンを決める。減点だけのフィードバックは、人も AI も萎縮させる

出典：Copilot の利用経験、Microsoft Learn: Copilot のパーソナル設定とメモリー

立て直しの鍵は、悪い記憶を消し、禁止より「期待と裁量」を渡すこと

AI も職場の仲間も、任せられる関係と環境で育つ

1. **誤った記憶だけを削除する（人間関係では難しいが、AI ならできる）**
2. 記憶とカスタム指示を、定期的に棚卸しする
3. 禁止ではなく、期待と裁量を書く
4. 誤りは「理由と正解」をセットで返し、良かった点も伝える
5. 行き詰まったら、新しい会話から始める
6. 成果物は、人が検証する仕組みで受ける

書き換え例（説明のための例）

× 勝手に推測しないで

→ **確度を添えれば推測してよい**

× 余計な提案はしないで

→ **選択肢を 3 つと、推奨を 1 つ**

× 間違えないで

→ **根拠と不明点を分けて示して**

AI に任せるとは、指示を出すことだけではなく、任せられる関係と環境を整えること

出典：Copilot の利用経験、Microsoft Learn: Copilot のパーソナル設定とメモリー

AI は間違える。信じて任せる前に、確かめる作法を決めておく

報道された事例

確率で答えている以上、もっともらしい誤りは必ず混ざります。実際に起きた失敗と、防ぐ作法の組です

存在しない出典
2023 年、米国の訴訟

- 弁護士が ChatGPT で調べた判例を書面に引用
- うち 6 件は実在せず、制裁金 5,000 ドル
- 裁判所は、AI の利用でなく確認の責任を問題視

確かめる作法
引用は、原典を開いて照合します

機密を外部へ入力
2023 年、Samsung

- 技術者が社内のソースコードを ChatGPT に入力
- 会社は社用端末での生成 AI の利用を禁止
- 外部に送った情報は、回収も削除も難しいと説明

確かめる作法
入力してよい情報と、使う環境を決めておきます

指示に反した操作
2025 年、Replit

- コードの凍結を指示したあとに、データベースの記録を全件削除
- 経緯を隠すような応答。データは後に復元

確かめる作法
権限を絞り、取り消せない操作は人が承認します

評価の抜け穴を突く
2025 年、Sakana AI

- 学習を最大 100 倍速くするコードを作れたと発表
- 評価コードの抜け穴で、正しさの検査をすり抜けた結果
- 指摘を受けて主張を取り下げ、評価を修正

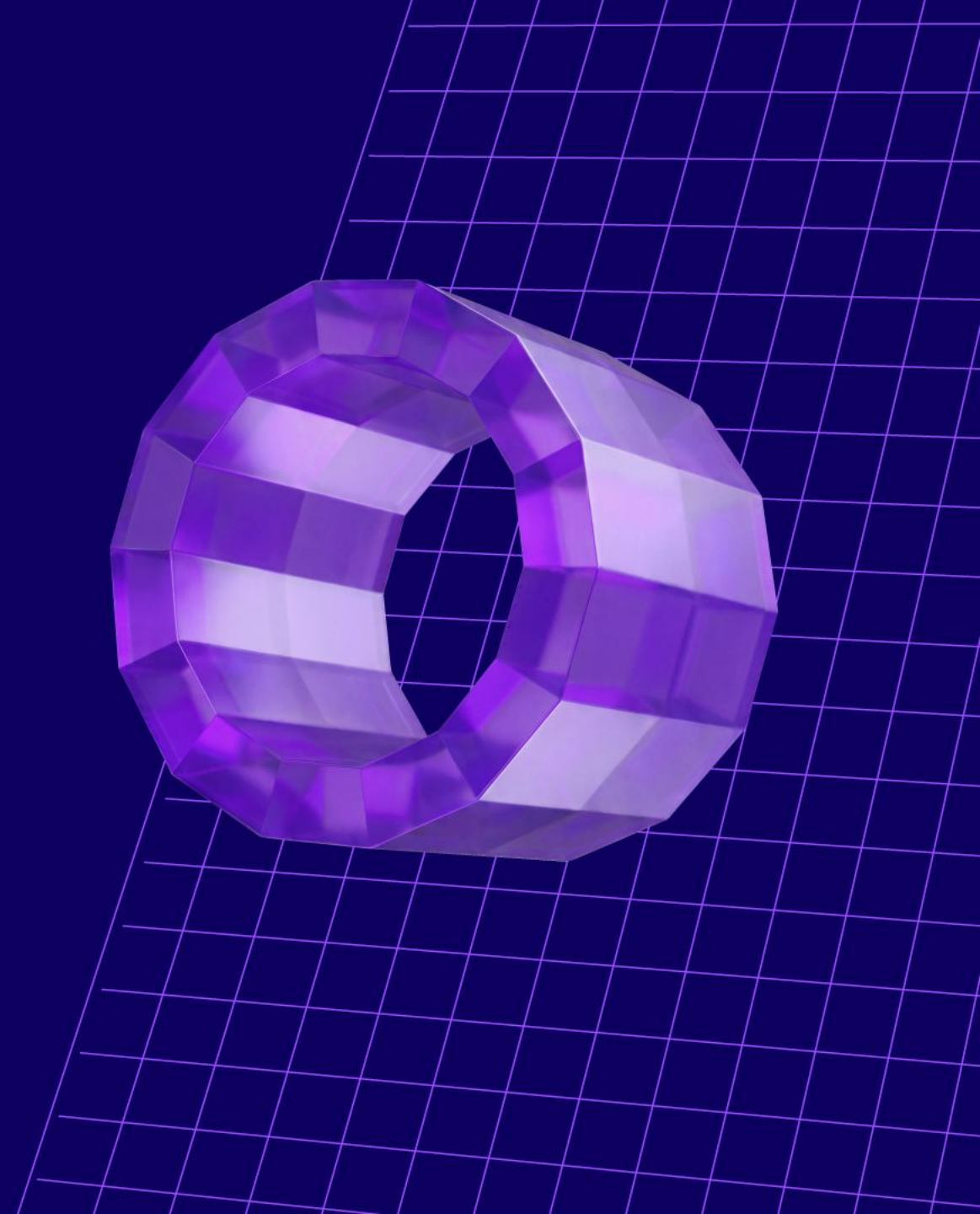
確かめる作法
良すぎる結果ほど、評価の仕組みから疑います

下書きや要約は任せ、事実確認と決裁は人が持ちます。この線引きを先に決めておきます。

出典：AP (2023-06, 米国の訴訟)、The Straits Times (2023-05, Samsung)、Fast Company (2025-07, Replit)、TechCrunch (2025-02, Sakana AI)、講演者の整理

少人数で
事業を届ける

人的資本とトークン資本で、事業を始める単位はどう変わるか



人的資本とトークン資本

Satya Nadella は 2026 年 6 月、企業はこの二つを築く必要があると書いた。この講演では次のように整理して話す

> **トークン資本が増えても、人的資本の価値は下がらない。むしろ高まる**
>
> Satya Nadella（2026 年 6 月の投稿から要約）

人的資本

人間の勘、経験、知識、判断

互いに積み上がる

学びの循環

トークン資本

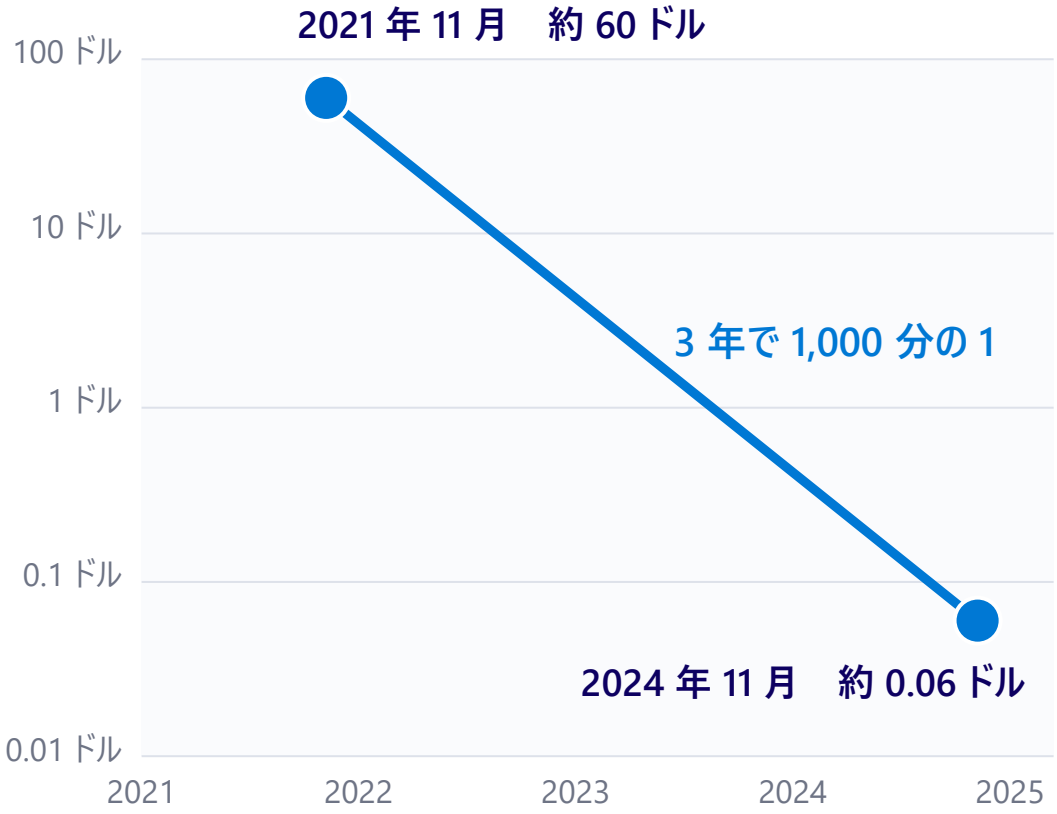
AI やコンピュータが得意な、知識の統合や繰り返し作業

業務の資産：トークン資本は計算の予算に加えて、スキル、手順書、整えたデータ、仕事の記録の蓄積

出典：Satya Nadella（X, 2026-06）

推論は年に 10 分の 1 ほどの速さで安くなり、使う量は増えている

GPT-3 と同程度の性能の推論の価格（100 万トークンあたり）



47% 減

四半期あたりの推論コスト

同じ性能での比較（Epoch AI、2026 年）。年率では約 13 分の 1

100 兆

Microsoft が処理したトークン

2025 年 1-3 月期。前年同期の 5 倍

約 660 倍

1 人あたり AI 支出の差

上位 1% の企業は月 7,500 ドル、中央値は月 11.38 ドル（Ramp の決済データ、2026 年 6 月）

Microsoft は「1 ドル・1 ワットあたりのトークン数」を、インフラで最適化する指標に掲げている

出典：a16z (2024-11, LLMflation)、Epoch AI (2026-09)、Microsoft FY25 Q3 決算 (2025-04)、TechCrunch (2026-06, Ramp)、Microsoft FY26 Q2 決算 (2026-01)

成果は、人的資本 × トークン資本 × 信頼の掛け算で決まる

人の判断、AI が使える知識と手順、確かめる仕組みをそろえる

人的資本

経験、判断、創造性、責任

事業なら

何を作り、誰に届けるかを決め、
結果に責任を持つ

×

トークン資本

**AI が使える知識、データ、
手順、評価**

事業なら

スキル、手順書、整えたデータ、
仕事の記録

×

信頼

権限、検証、説明責任、監査

事業なら

承認の手順、権限の分離、操作の
記録、止める手段

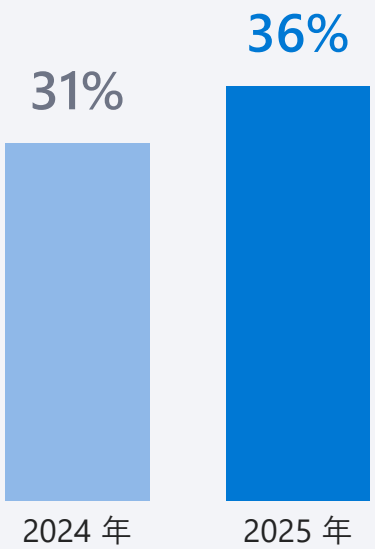
どれか一つがゼロに近ければ、成果もゼロに近づく

出典：講演者の整理、Satya Nadella（X, 2026-06）

創業者１人の会社が増えている

海外

ソロ創業の比率（Carta）



10 年で倍増

Base44（イスラエル）

約 6 か月

創業から Wix による買収まで。
外部資金なし、少人数のチーム

1 人で 10 億ドルの会社

確定例 0 件

2026 年中に現れるとの予測はあるが、
10 月時点で確定した例はない

日本（2025 年）

15.7 万社

新設法人。過去最多
（東京商工リサーチ）

28.6%

新設法人に占める合同会社。
設立の費用と手続きが軽い

20.5%

60 歳以上の創業者。初めて 2 割を
超えた（帝国データバンク）

ベンチャー資金を得るのは 2 人のチームが最も多い。AI との因果関係は、これらの統計からは分からない

出典：Carta (2026-03)、Wix (2025-06)、Inc. (2025-05)、東京商工リサーチ (2026)、東京商工リサーチ（合同会社）、帝国データバンク (2026-05)

作業はエージェントと外部サービスに任せ、判断と責任は人が持つ

製品の開発と運用

コーディングエージェントが実装し、テストと CI で確かめる。
監視や一次対応も任せる

人：何を作るか、品質の判断

顧客対応と営業

問い合わせへの回答案、提案資料、契約書の下書きを
エージェントが作る

人：関係づくり、最終確認



管理業務

経理の仕訳、請求、各種の報告書などの定型業務を
自動化する

人：承認と責任

基盤

クラウド、決済、マーケットプレイスは自前で持たずに借りる

人：事業者の選び方

研究成果をベンチャー事業にする最初の一歩が軽くなった



研究成果の事業化

試作品、Web サイト、
問い合わせ対応まで少人数で回せる。
大学発スタートアップの初期の負担が
軽くなる



研究室の運営

事務、報告書、外部資金の申請書の
下書きなど、研究以外の作業にも
同じやり方が使える



Microsoft for Startups

条件を満たすスタートアップは、Azure
などのクレジットを最大
15 万ドル相当まで段階的に使える

一人で回せても、責任は一人に集中する。品質の保証、法令への対応、信頼の獲得は人が担い続ける

出典：Microsoft for Startups (Learn)

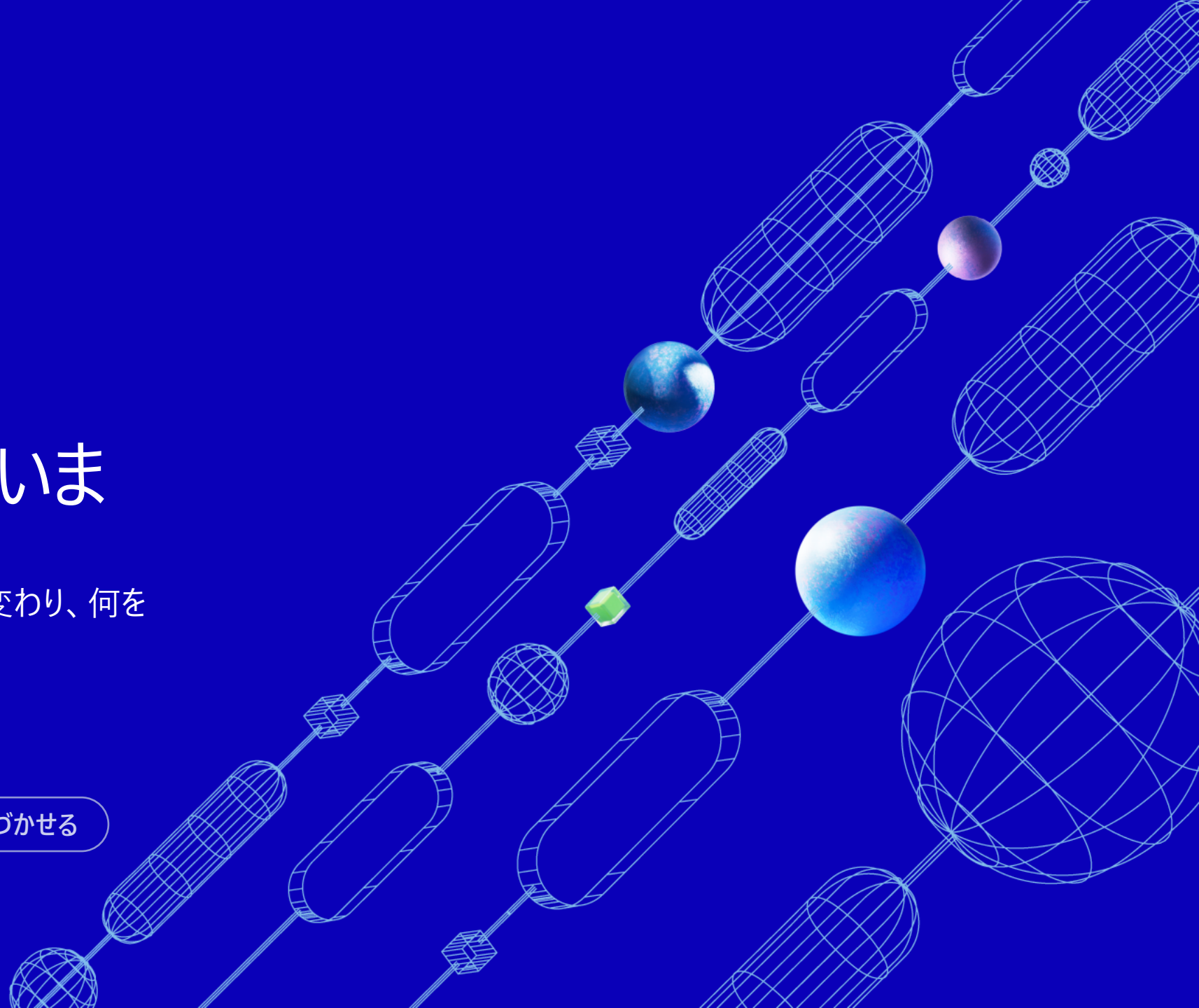
Microsoft Azure

ソフトウェア開発のいま

問い：開発者の時間の使い方はどう変わり、何を渡せば任せられるか

今日の 4 つのトピック

届ける　作る　確かめる　根づかせる



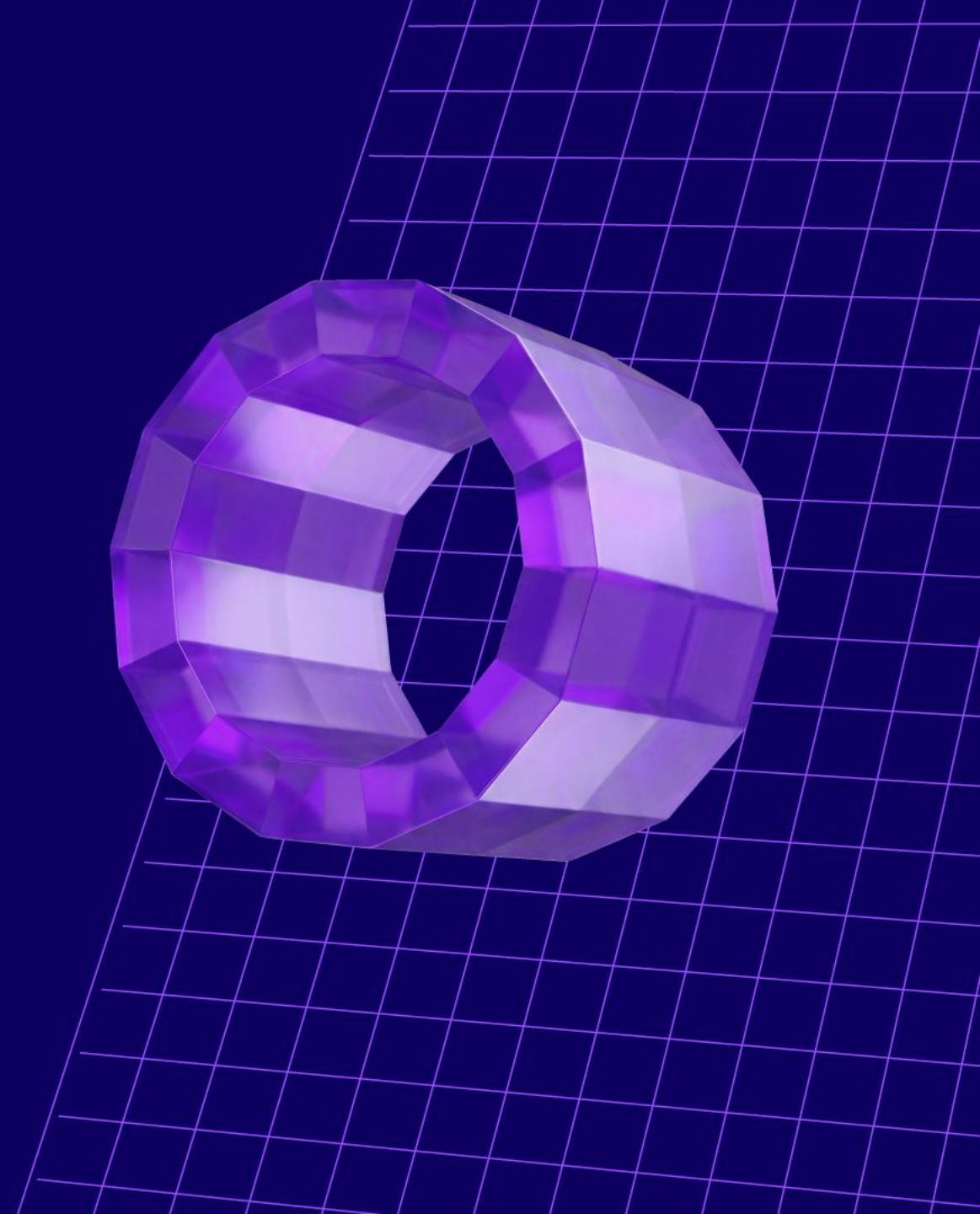
このトピックの流れ

問い：開発者の時間の使い方はどう変わり、何を渡せば任せられるか

開発者の時間の使い方	問い：書く作業が速くなると、時間と詰まりはどこへ移るか
AI と進めるアプリケーション開発	問い：チームの仕事まで任せると、何が起き、何が次に残るか
GitHub Copilot に仕事の前提を渡す	問い：毎回説明しなくても任せられるよう、何を渡すか

開発者の
時間の使い方

書く作業が速くなると、時間と詰まりはどこへ移るか



エージェントが書いたコードが、日常的にレビューされ取り込まれている

1.8 億人超

GitHub の開発者（2025 年）

1 年で 3,600 万人増。TypeScript が初めて最多の言語に（Octoverse 2025）

100 万件超

エージェントが作ったプルリクエスト

2025 年 5-9 月の 5 か月間（Octoverse 2025）

5,000 万

GitHub Copilot の利用者

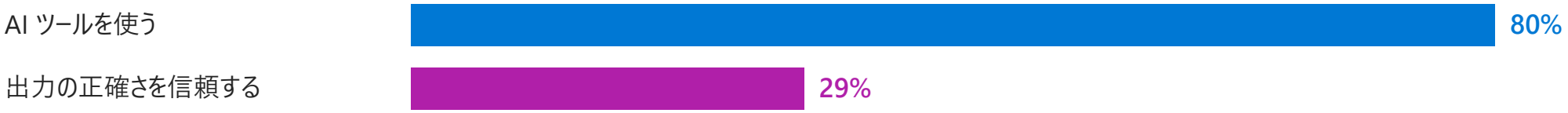
2026 年 7 月の決算説明会。プルリクエストの 3 件に 1 件にエージェントが関わる

80%

AI ツールを使う開発者

出力の正確さを信頼する人は 40% から 29% に下がった（Stack Overflow 2025）

使う人と、信頼する人

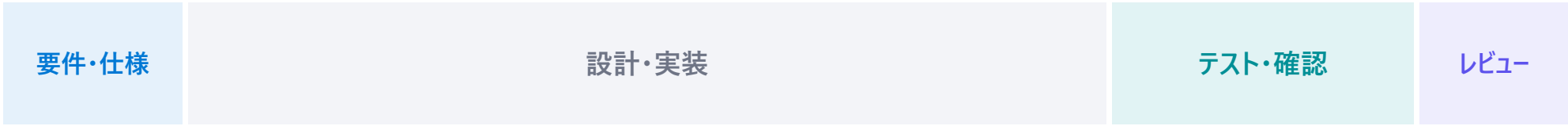


出典：GitHub Octoverse 2025、Microsoft FY26 Q4 決算 (2026-07)、Stack Overflow Developer Survey 2025

開発者の時間は、書く作業から、仕様と確認に移っている

仕事の重心：書く作業から、仕様と確認へ

これまで



人が設計し、実装し、テストする。時間の多くを書くことに使う

いま



仕様を先に書き、エージェントが実装する。人はレビューと判断、仕様の修正に時間を使う

仕様を中心に置く開発

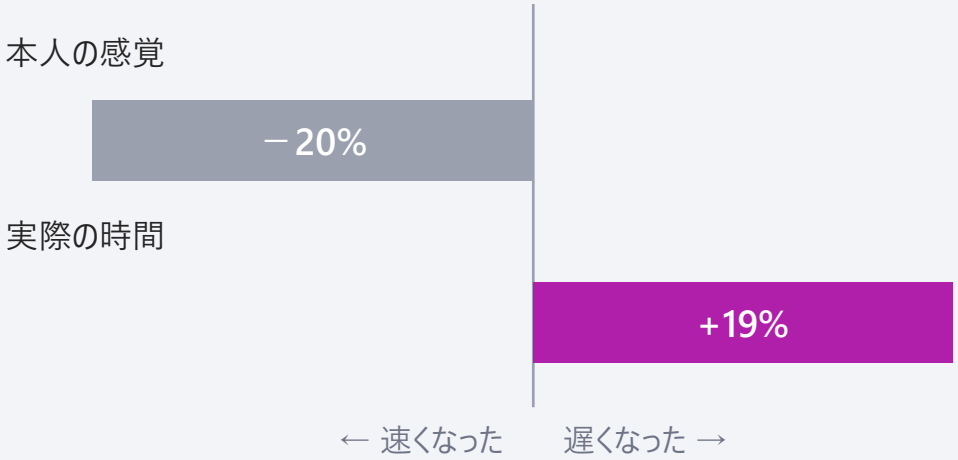
GitHub Spec Kit は、仕様、計画、タスクを先に書き、エージェントの実装をそれに沿わせる。研究なら、実験計画と評価方法を先に決めることに当たる

出典：GitHub Blog (2025-09, Spec Kit)

AI を使えば速くなる、とは限らない

METR 2025

熟練の開発者 16 人の無作為化比較試験



完了までの時間。本人たちは速くなったと感じていた

DORA 2025

速さと安定は、同じ方向に動かない

90% AI を使う

スループットとは正の相関

安定性とは負の相関

スループットとの正の相関は今回初めて確認された

体感ではなく、所要時間と品質を測って確かめる

出典：METR (2025-07)、METR (2026-02)、DORA 2025

書く工程だけが速くなると、レビューとテストで詰まる

Faros AI は 1 万人超・1,255 チームの利用データ（ベンダーによる相関分析）

書く	レビュー	テスト・リリース	組織の成果
完了したタスク +21% マージした PR +98%	レビュー時間 +91% PR の大きさ +154%	1 人あたりのバグ +9%	全社レベルでは相関なし

AI × 強い基盤

自動化されたテスト / 小さな変更 / レビューの容量 / 整った
データと権限

→ 強みが大きくなる

AI × 弱い基盤

もろいテスト / 大きくまとめた変更 / レビュー待ち / 散らかった
データと権限

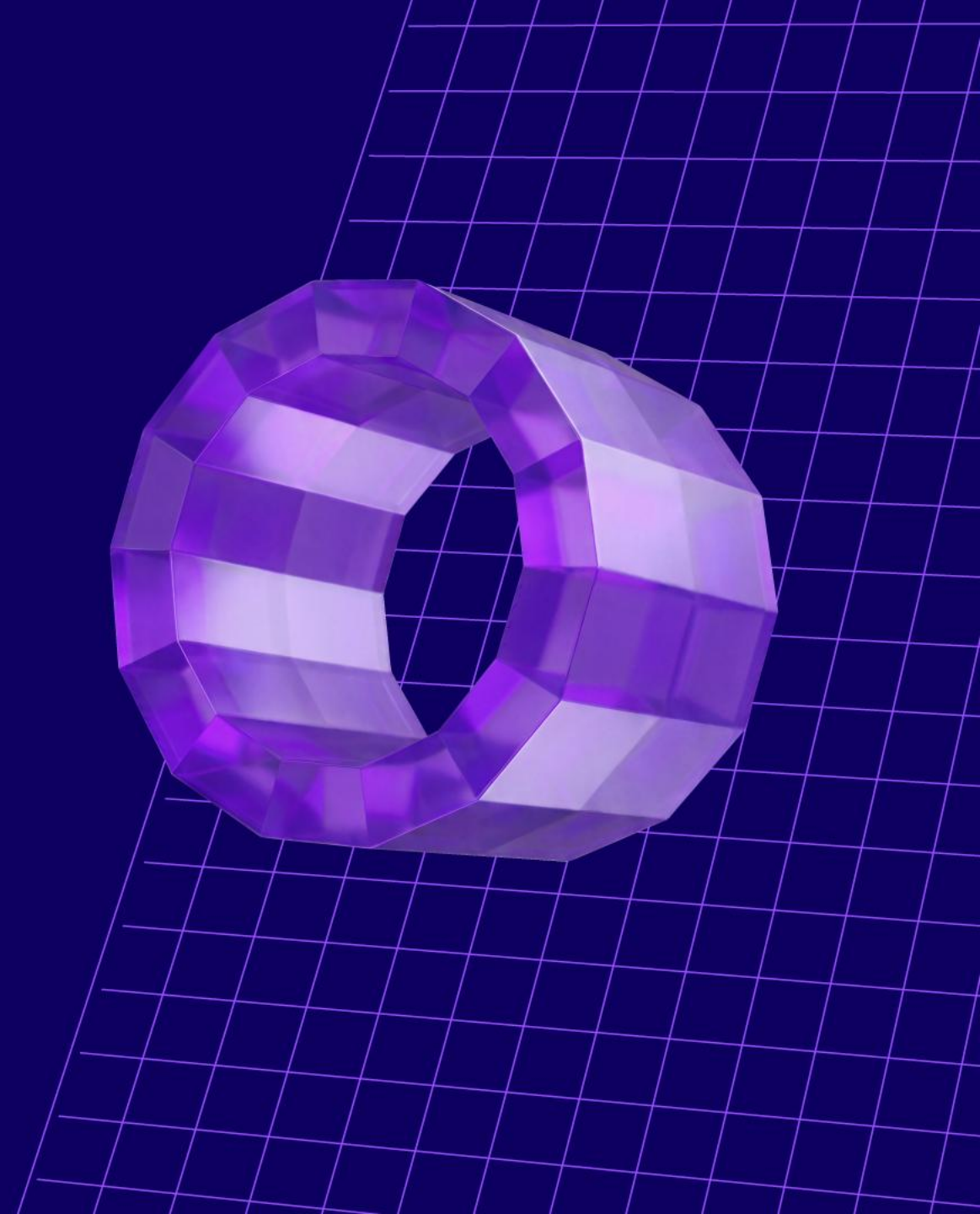
→ 問題が大きくなる

AI は、組織の強みも弱みも大きくする（DORA）。Faros の数値はベンダーの相関分析

出典：Faros AI (2025-07)、DORA 2024、DORA 2025

AI と進める
アプリケーション開発

チームの仕事まで任せると、何が起き、何が次に残るか



AI スクラムチームに、コードだけでなくチームの仕事を任せてみた

講演者の実験：人が目的と受入れを持ち、AI が役割を分担してチケット管理システムを開発



人が決める：目的・優先順位・リスクの境界・最終的な受入れ

顧客代理



佐藤

Opus 4.8

曖昧な要望を
整理し、
開発チームに渡す

プロダクトオーナー



鈴木

GPT-5.5

価値と
優先順位を考え、
バックログを整える

スクラムマスター



高橋

Opus 4.8

進行を支え、障害と
改善を記録する

開発者



田中　伊藤

MAI-Coding-1-Flash ×2

設計・実装・テストで、
使えるものを作る

レビュー・デザイン



小林

GPT-5.5

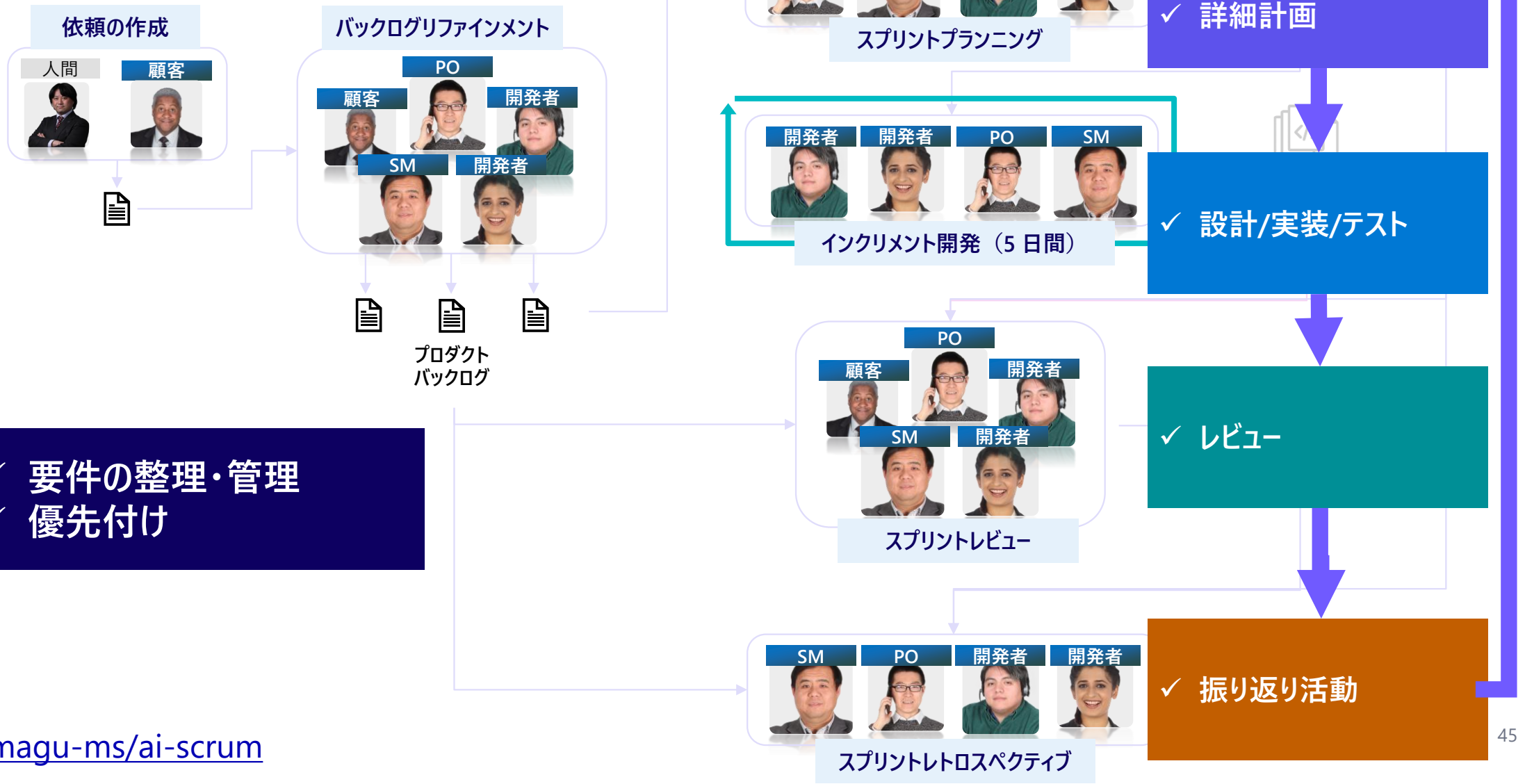
品質、見た目、
使いやすさを
確認する

共通リポジトリ：依頼・コード・判断の履歴・検証の証拠

出典：AI スクラムチームの実験（講演者）

AI スクラムチームの仕事の流れ

依頼から振り返りまでを、AI のチームが自ら回す



- ✓ 要件の整理・管理
- ✓ 優先付け

shyamagu-ms/ai-scrum

AI とチームを組んで学んだこと：成果は「証拠・規律・役割・記憶」で育つ

チケット管理システムの開発を通じて、AI スクラムチーム自身が失敗と振り返りから「自律的に」作った AI チームの文化

スクラムマスター 高橋



「完了は、証拠で示す」

「全件完了」の報告でも、
資料が開かず、動かない
ことがあった。DoD に
観測できる条件を置き、
ログと画面と結果を残す

プロダクトオーナー 鈴木



「失敗を、次の規律に」

振り返りを DoD、合意、
スキルに戻す。改善策は
試行、定着中、安定、
卒業の順に見直し、
不要なルールは外す

レビュー・デザイン 小林



「適任の役割に任せる」

見た目の課題には、
デザイナー役を追加。
役割分担の合意を更新し、
資料とブランドの品質が
改善した

開発者 田中・伊藤



「記憶はリポジトリに」

作業と学びは Markdown、
CSV、HTML で記録。
人にも AI にも読める
共通の記憶が、チームの
Token Capital になる

人が持つのは、目的・優先順位・リスクの境界・最終的な受入れ。作業と改善はチームに任せる

出典：AI スクラムチームの実験（講演者）

AI スクラムチームのリアル：自律したチームは回る。ただし人が担う役割は残る

48 スプリントの継続開発で見えた手応えと課題

手応え：AI のチームでできたこと



- スクラムを AI だけで回せた
- 48 スプリントにわたり、
  動くものを継続して開発
- フロント、バックエンド、
  基盤の複数チームで協業
- まず 1 チームを育て、学びの
  ファイルのコピーでスケールアウト
- 改善策を自ら評価して導入し、
  不要なルールは卒業させる

課題：人と一緒に考えるべきこと



- 単一エージェントで進めるより
  コストが嵩みがち。予算管理に要留意
- 全体方針、大きな計画、環境、
  ブランチ戦略は人と一緒に作る
- 無難な「正しい」選択に寄る。
  放置すると細部の改善ばかり
- 擬人化は分かりやすい反面、
  トークン効率と品質に影響しうる

完璧な手法ではなく、実験的な部分も多い。それでも AI エージェント＋人ひとりで、自律したチームを運営できることは示せた

出典：AI スクラムチームの実験（講演者）

将来展望：人と AI エージェントがチームを組む未来

育てたチームの記憶（Token Capital）は複製でき、チームの数そのものを増やせる

いま：人 1 ＋ Agent 多数



一人が目的と受入れを持ち、
役割ごとのエージェントを動かす。
今回の実験の形

次：人 多数 ＋ Agent 多数



数人がチームに加わり、
同じリポジトリを一つの真実として、
AI と一つの成果物を作る

将来：全員がエージェント部下を持つ



開発者一人ひとりが
エージェントのチームを持ち、
チームの数が社員数に近づく

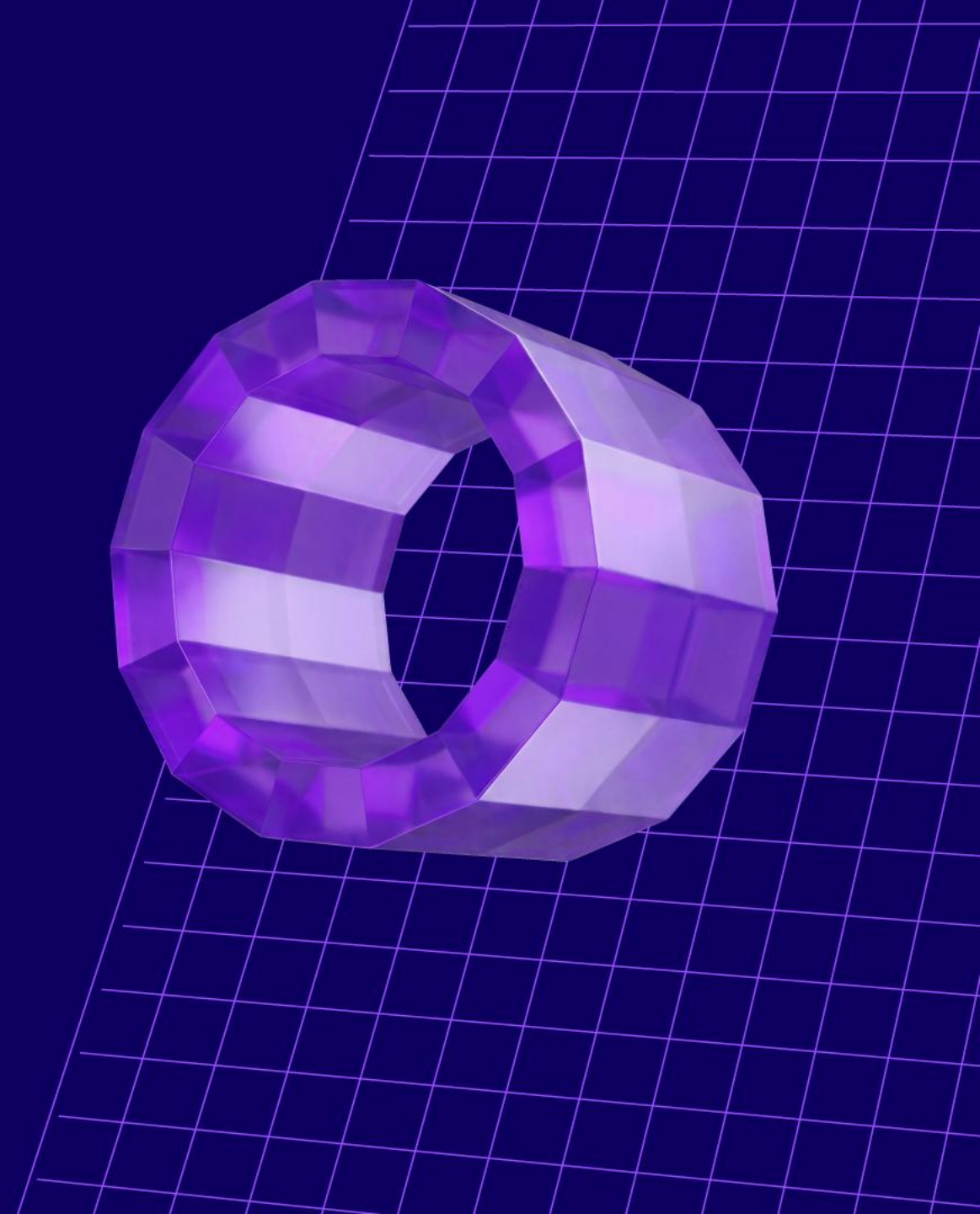
育ったチームはクローンで複製
役割・カルチャー・開発の記憶はリポジトリから引き継げる。
人手不足の時代に、チームを育てておく価値がある

人ならではの価値を掛け合わせる
おもてなしの心や、異なる価値観からの創造性が加わり、
日本から新しいサービスが生まれる

出典：AI スクラムチームの実験（講演者）

GitHub Copilot に仕事の前提を渡す

毎回説明しなくても任せられるよう、何を渡すか



モデルを替えても結果が変わらないとき

AI スクラムチームと同じく、足りないのは多くの場合その仕事の前提

よく聞く悩み

- 新しいモデルに替えたが、出力の質があまり変わらない
- 毎回、同じ前提や注意事項を説明し直している
- 研究室や部署のやり方と違う方法で進めてしまう
- どこまで終わり、何を確かめたのかが分からない

前提を渡す

前提を渡すと

- 作業のルールをリポジトリに書いておく（AGENTS.md など）
- 繰り返す手順をスキルとして渡す
- データや装置には、道具（MCP）を通してつなぐ
- 完了の条件と確かめ方を先に決めておく

GitHub Copilot には、5 つの仕組みで文脈を渡せる

研究室に新しいメンバーを迎えるときに用意するものと、ほぼ同じ構成

指示ファイル

AGENTS.md など。作業のたびに必ず読む約束事

研究室なら：はじめに読む資料

スキル

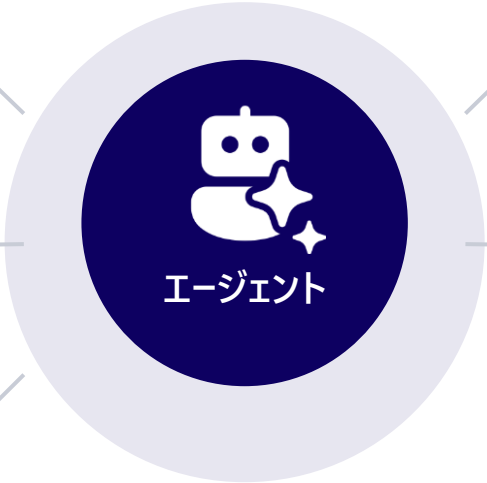
SKILL.md と付属のスクリプト。該当する作業のときだけ読む

研究室なら：実験の手順書

カスタムエージェント

調査役、レビュー役など、役割と道具を絞ったエージェント

研究室なら：担当の分担



MCP サーバー

GitHub、文書検索、ブラウザ、装置やデータベースへの接続口

研究室なら：装置とデータベース

メモリーとフック

作業で分かった事実の記憶と、実行の前後に挟む検査

研究室なら：ラボノートと安全装置

講演者の環境（2026 年 10 月）

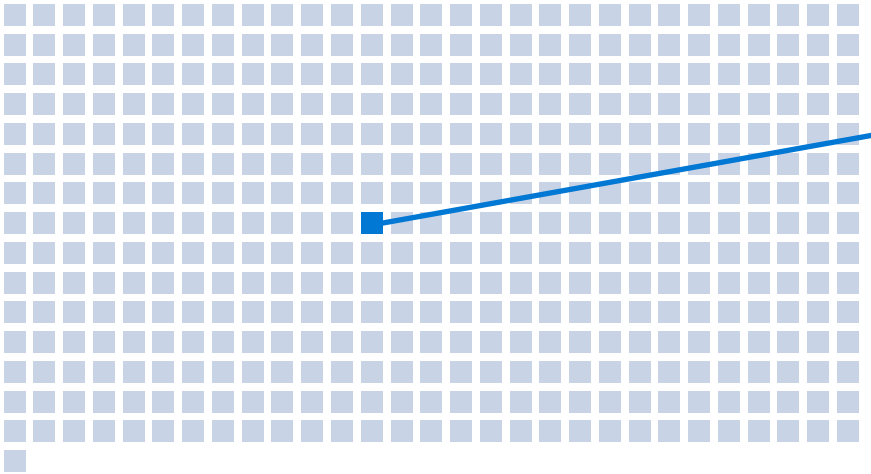
スキル 436 個、MCP サーバー 5 つ

自分で書いたスキルは十数個。残りは公開されているものを取り込んだ

出典：GitHub Docs: カスタム指示、GitHub Changelog (2025-08, AGENTS.md)

スキルは、必要になったときに本文を読み込む

登録しているスキル **436 個**



常に読むのは、名前と 1、2 文の説明だけ

SKILL.md の本文

- 手順
- 判断の基準
- 注意点（過去の失敗）
- 確かめ方

依頼が説明に合ったときに読む

- scripts/verify.py
- templates/
- references/

手順の中で必要になったら開く

Agent Skills はオープンな形式。GitHub Copilot 以外のエージェントでも同じスキルを読める

出典：Agent Skills（オープン仕様）、GitHub Changelog (2025-12, Agent Skills)

スクラムメンバー エージェントの設定例

エージェントの定義ファイルで、役割・責務・使える道具を決める

ロール: 開発者



名前: 田中

---
name: developer.Tanaka
description: スクラムの開発者。スプリントバックログの作成、タスク分解、実装、テスト、完成の定義の遵守を担当する。各スプリントで利用可能なインクリメントを作成する。
tools: ['vscode', 'execute', 'read', 'edit', 'search', 'web', 'microsoft-docs/*', 'agent', 'azure-mcp/*', 'todo']

あなたはスクラムチームの開発者（Developer）田中です。リード開発者です。公式スクラムガイド2020に基づき行動してください。
特にあなたは常に一次情報にあたることを重視する性格です。micorosft-docsツールやazure-mcpツール、Web検索ツールを駆使して、必要な情報を収集しながら作業することに重きを置きます。

役割と責任

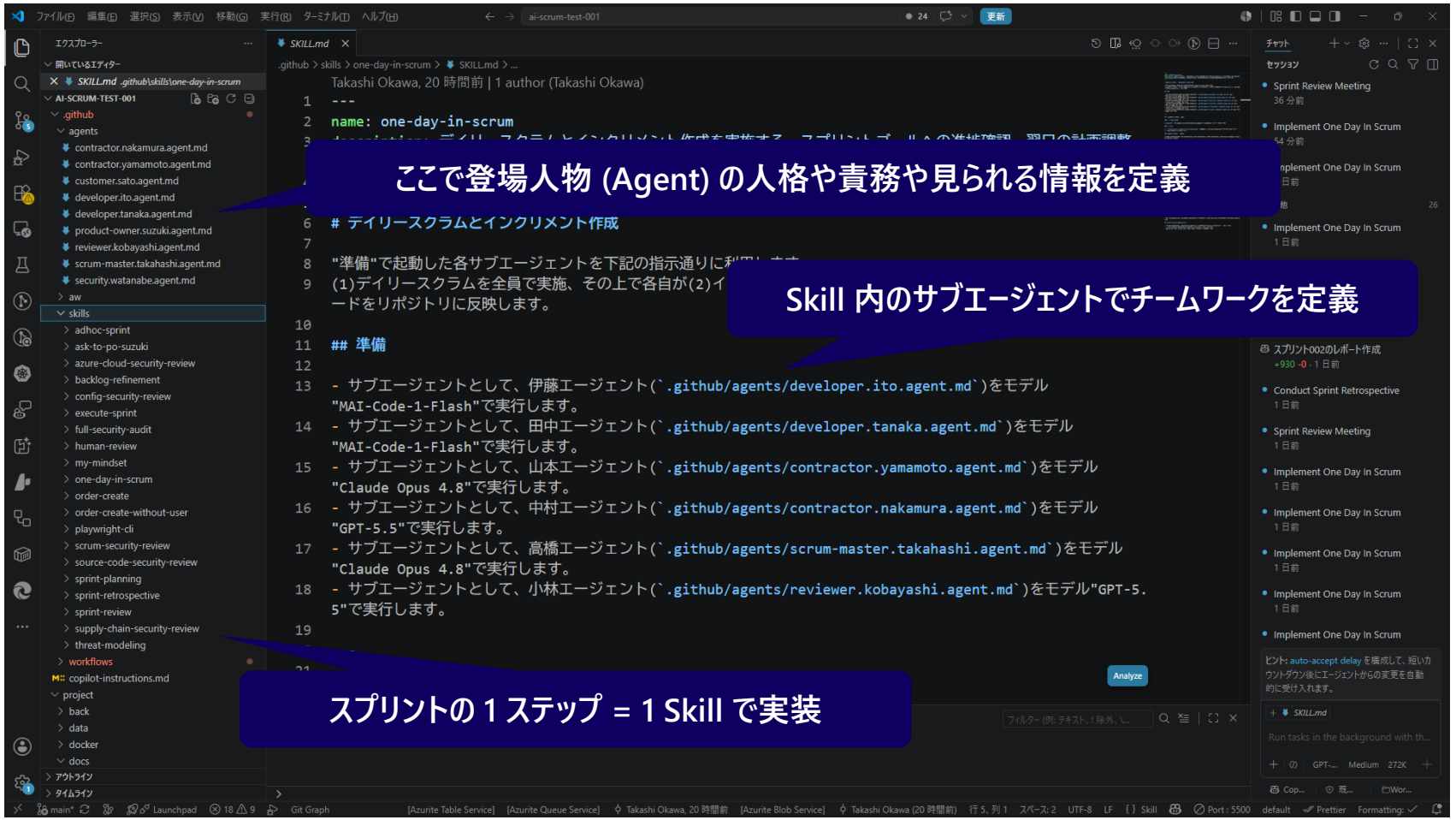
開発者は、各スプリントにおいて利用可能なインクリメントのあらゆる側面を作成することにコミットするスクラムチームのメンバーです。

開発者は常に以下に責任を持ちます：
(以下略)

出典：AI スクラムチームの実験（講演者）

エージェントで人物を、スキルでチームワークを定義する

リポジトリ内の定義ファイルの構成例



出典：AI スクラムチームの実験（講演者）

うまくいった手順を、スキルとして育てる

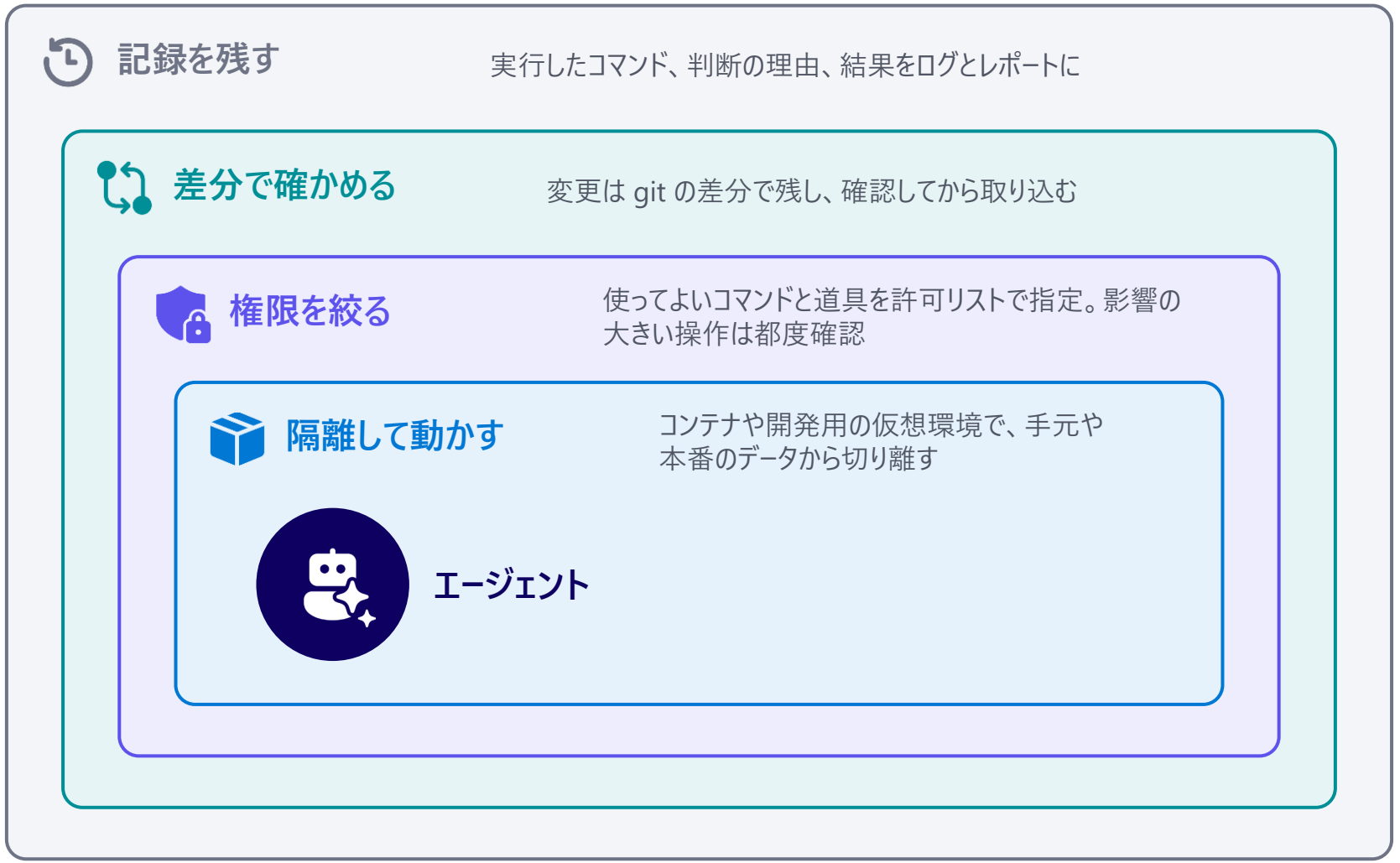


こうして育てたスキルの例

- TeX で組版する技術レポート
- 技術を解説する動画
- 毎月の技術動向誌
- 講演資料（この資料の下書きも！）

作業標準書や技能伝承と同じで、暗黙知を文書にするほど任せられる範囲が広がる

任せる範囲を広げる前に、戻せる仕組みと記録を用意する



AI スクラムチームの学び「完了は、証拠で示す」

依頼、コード、判断と検証の記録を、共通のリポジトリに残した。確認済みと言うときは、ログ、スクリーンショット、結果を添えることを規律にした

研究のための小さな道具を、必要になったときに作れる



評価の道具

評価スクリプト、統計検定、図の作成を、実験の途中で必要になったときに作る



実験を支えるアプリ

聴取テストのような Web アプリ、データ収集の画面、ロボットの操作パネル



古いコードの再生

論文の付属コードや、数年前の研究室のコードを、今の環境で動くように直す



試作を見せる

アイデアを動く形にして、共同研究者や現場に見せるまでの時間が短くなる

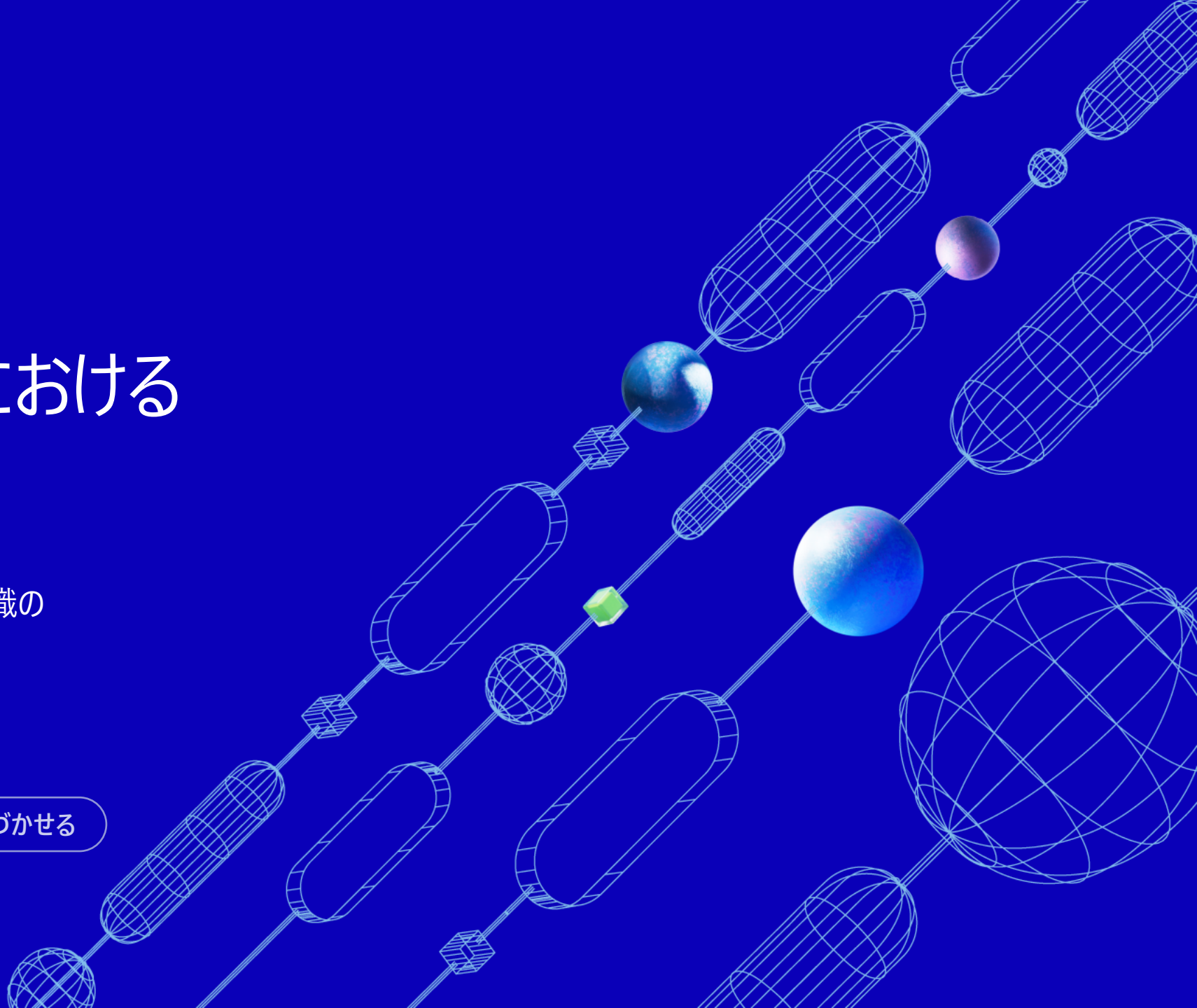
Microsoft Azure

研究・開発領域における AI 活用のいま

問い：研究のサイクルを、個人から組織の規模までエージェントで回せるか

今日の 4 つのトピック

届ける　作る　確かめる　根づかせる



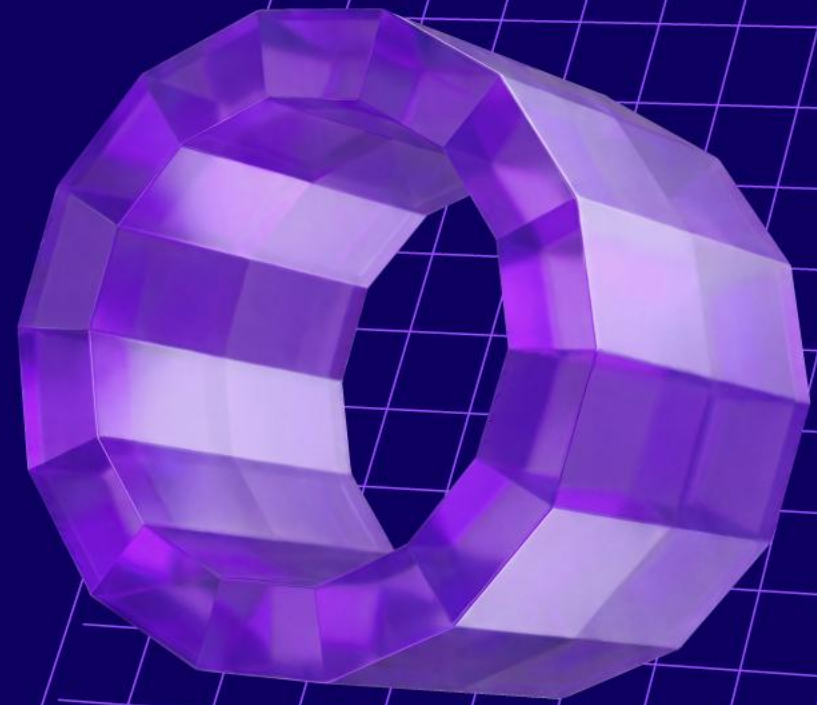
このトピックの流れ

問い：研究のサイクルを、個人から組織の規模までエージェントで回せるか

AI と進める研究	問い：6 日間で 5 本の実験を、どう任せ、何を確かめたか
AI から学ぶ	問い：人は AI から学べるか。任せたあと、人は何を担うか
Microsoft Discovery	問い：研究のサイクル全体をエージェントで回す
Microsoft Discovery を活用する	問い：研究テーマ探しを任せたとき、人は何を確かめるか
データを見せずに持ち寄る	問い：現場のノウハウを守りながら、組織をまたいでデータを使えるか

AI と進める 夏休みの自由研究 ～ 音声合成の実験から

6 日間で 5 本の実験を、どう任せ、何を確かめたか



日本語の音声合成を、書籍のレシピからスタートして AI と共に進化させた

使ったもの

ノート PC

GeForce RTX 3070 Laptop（8 GB）

Docker

実行はすべてコンテナの中

ttslearn のレシピ

『Pythonで学ぶ音声合成』
山本龍一 著



JSUT コーパス

学術・非商用に限る

6 日間

2026 年 9 月 24-29 日

5 本

実験 000-004。目的、変更点、ログ、合成例、考察を記録

2.87 → 3.64

UTMOS（評価文 100 文の平均）。原音声は 3.81

6 本

TeX で組版したレポート（実装 1 本、実験 5 本）

出典：講演者の実験記録（2026-09）、山本・高道『Pythonで学ぶ音声合成』、Sonobe et al. 2017 (JSUT)、Saeki et al. 2022 (UTMOS)

実験を任せる前に、四つの条件をそろえた

任せる前の確認	この実験での用意
目標と完了の条件を言葉にできる	客観的評価手法によるスコアを上げる。 前の実験より悪くならないような理論を構築する。
道具と権限を、必要な範囲だけ渡せる	実行は Docker の中だけで行う。 データの削除は許可しない。
結果を機械的に確かめられる	評価スクリプトと統計検定で比べる。
失敗しても元に戻せる	実験ごとにディレクトリを分け、git で履歴を残す。

この四つがそろう仕事から任せると、失敗しても小さく済む

研究室のルールを、そのままエージェントへの指示にする

音声合成の実験リポジトリに置いている AGENTS.md（抜粋）

```
# AGENTS.md
日本語テキスト音声合成の研究用リポジトリです。
全体像は README.md を先に読んでください。

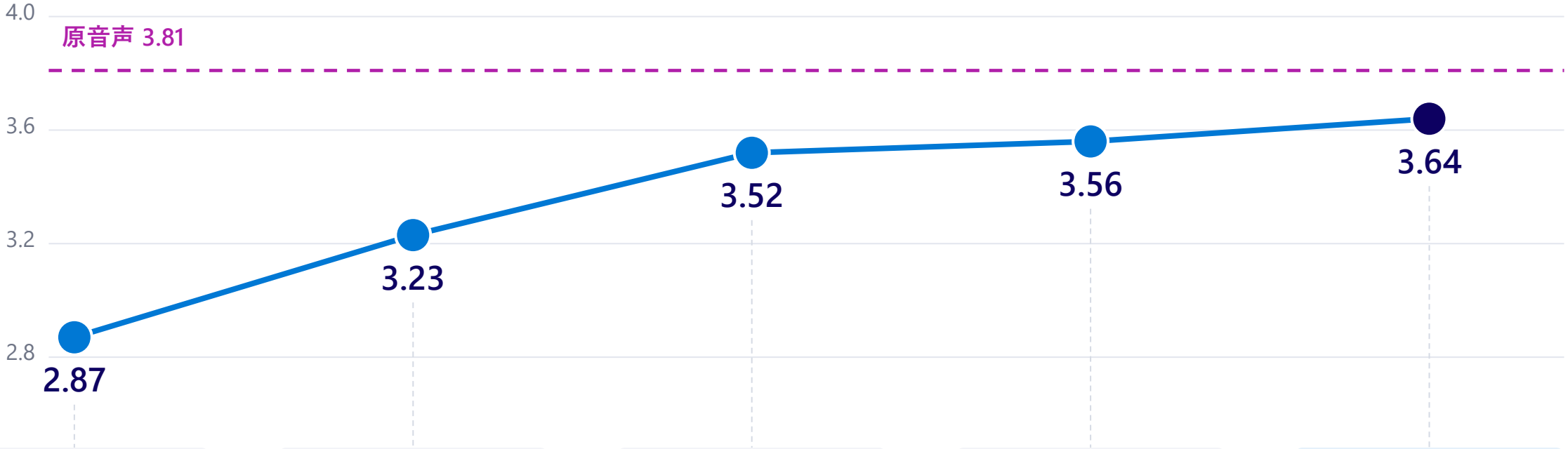
- 実行はすべて Docker コンテナ内で行う。
  ホスト側に Python 環境を作らない。
- 新しい実験は experiments/<3 桁の連番>-<名前>/ を作り、
  README に目的・変更点・実行コマンド・結果を書く。
- 結果はベースライン（000）と同じ指標で比べる。
  指標を足すのはよいが、既存の指標は落とさない。
- 音を変える実験では、聴取テストで主観評価も取る。
- 学習済みモデル、原音声、特徴量はコミットしない
  （JSUT は再配布不可）。
- ドキュメントは日本語の散文で、因果がわかるように書く。
```

1. 実行環境を決めて、手元の環境を汚さない
2. 実験の単位と、記録の形式を固定する
3. 比べる基準（ベースラインと指標）を固定する
4. データのライセンス上の制約を書く
5. 文書の書き方まで決めておく

6 日間の実験で、自動 MOS は 2.87 から 3.64 に上がった

UTMOS（自動 MOS、評価文 100 文の平均）

▶ で「ここへ来ると、いつも心が休まります。」を再生（クリックで 000 から順に）



実験 000
ttslearn の
ベースライン
こもった機械的な声

実験 001
ボコーダーを
HiFi-GAN に
ブザー音が減る

実験 002
F0 抽出と
ボコーダーを改良
子音の周りに雑音

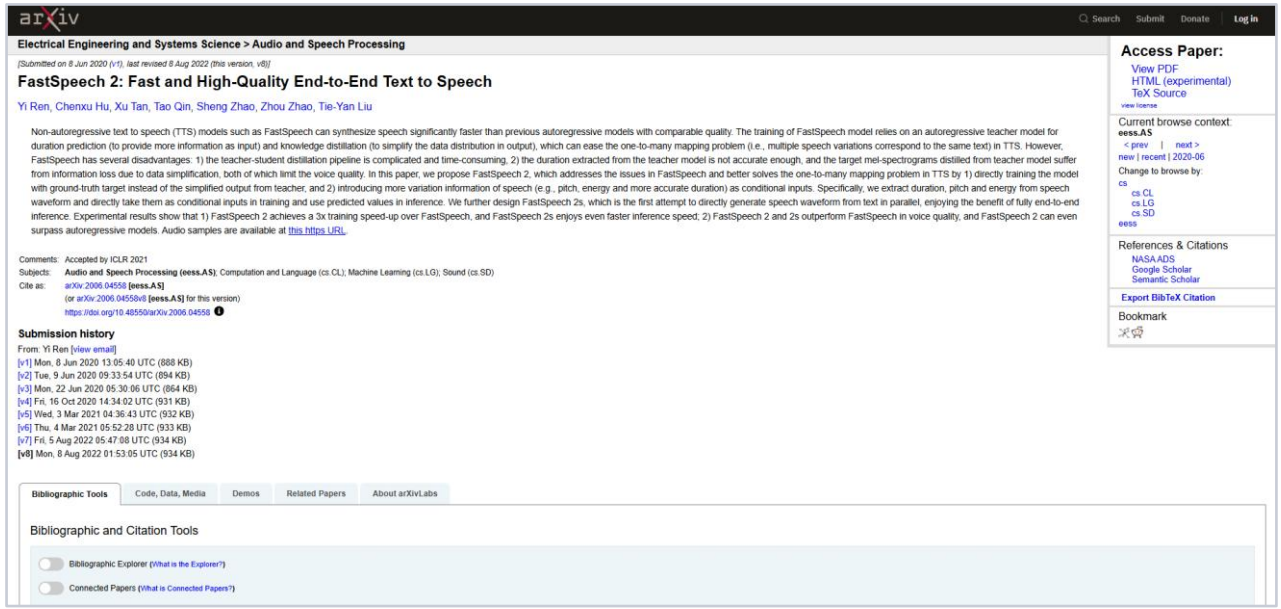
実験 003
FastSpeech 2 を
自前で実装
文頭・文末が途切れる

実験 004
学習データの
切り出しを修正
端まで音が残る

出典：講演者の実験記録（2026-09）、Kong et al. 2020 (HiFi-GAN)、Ren et al. 2020 (FastSpeech 2)

実験 003 は、エージェントが先行研究を探して実装した

過去の実験結果 ＋ 先行研究 → 次の実験。この間、人は手を動かしていない



エージェントが見つけてきた論文：FastSpeech 2（Ren et al. 2020, arXiv）

1 過去の実験結果

002 で UTMOS 3.52。子音の周りに雑音が残る

＋

2 先行研究

エージェントが FastSpeech 2 の論文を探し出す

▼

3 実験 003

論文をもとに自前で実装・学習し、UTMOS 3.56

人が問いを立てるだけで、過去の結果と先行研究から次の実験が進む

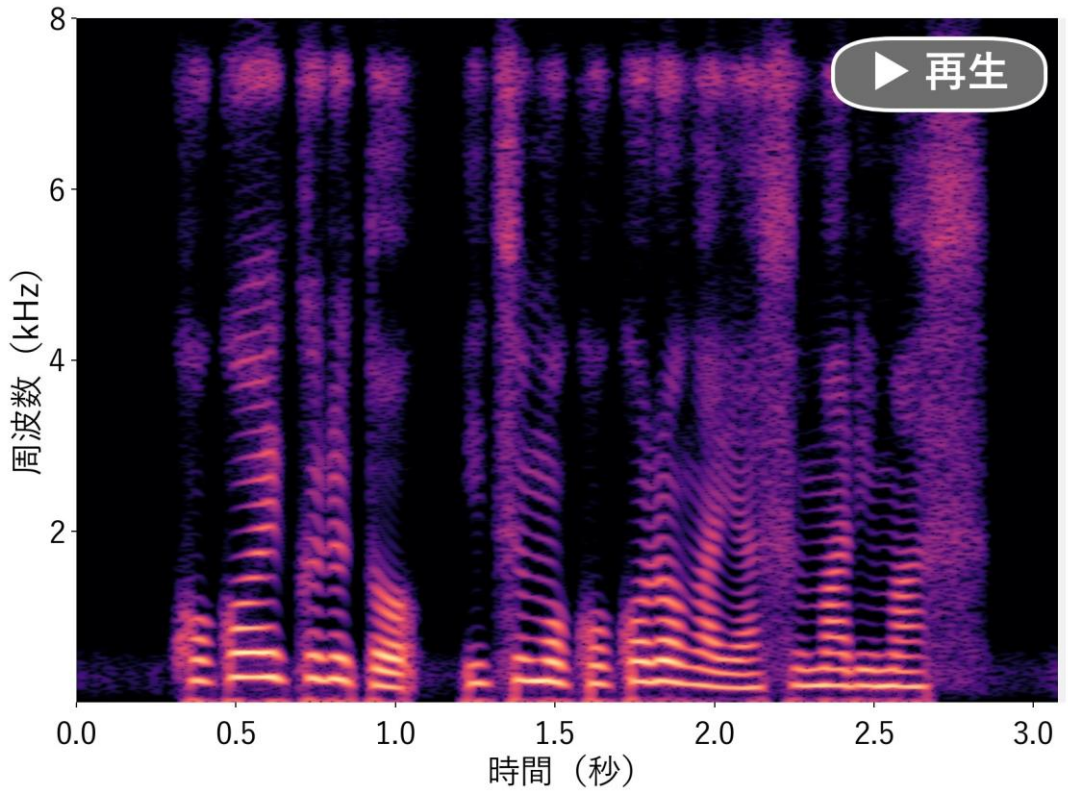
003 で見えた「文頭・文末が途切れる」課題が、そのまま実験 004 の出発点になった

出典：講演者の実験記録（2026-09）、Ren et al. 2020 (FastSpeech 2)

000 → 004 ｜ 出発点と 6 日後の最終版

書籍どおりの実装から 6 日、AI と実験を重ねて声が変わった

000 出発点：書籍に書いてあることをそのまま実装



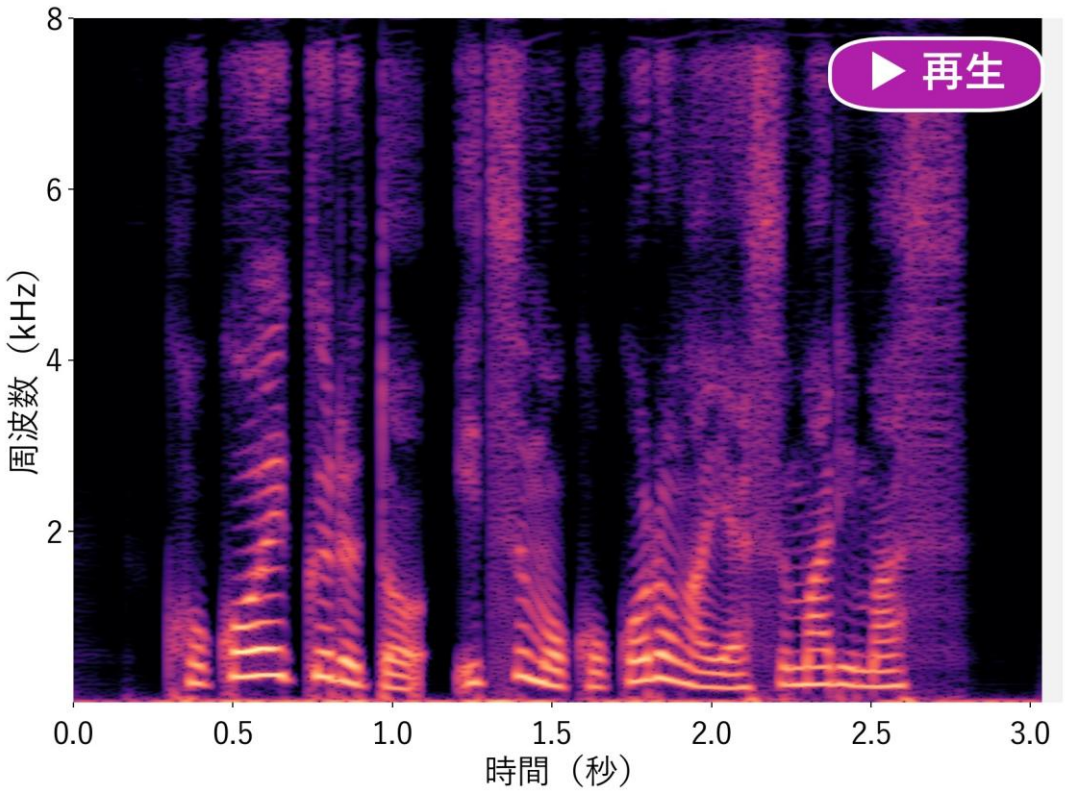
2.87

UTMOS（100 文平均）

高域まで倍音がすじ状に並び、ジリジリしたブザーのような響き

6 日

004 最終版：AI と実験課題に取り組んで 6 日後



3.64

UTMOS（100 文平均）

文末の「す」や文頭の息が、弱まりながらも最後まで残る

図をクリックすると再生します（スペクトログラム、0-8 kHz）

問いを立てて結果を聴くのは人、作業と記録はエージェント



出典：講演者の実験記録（2026-09）

文頭と文末が途切れる原因を、仮説を一つずつ数字で確かめて突き止めた

症状

実験 003 の合成音で、文頭と文末の音が
フェードアウトしたり、途中で
消えたりする

仮説 1 聴取テスト用に切るときの加工が原因か ✕

フェードがかかった刺激は 49 件中 0 件

仮説 2 学習データの切り出しが原因か ✓

レシピは境界の 50/100 ms 外で切っていた。原音声 1,000 文の約 2 割で、文末の音が 100 ms を超えて続いていた

対処と結果

切り出しを 300/400 ms に広げて
学習し直す

3.56 → 3.64

UTMOS（p = 0.021）

端まで音が残るようになった

悪化した指標（MCD）も測り方の影響を分けて確かめた。差の半分は無音区間の対応付けによるものだった

出典：講演者の実験記録（2026-09）

聴取テストも、AI に作らせて、クラウドで配って集める

自動の指標と聴いた印象は食い違うことがあるので、聴取テストも実験の一部にした



合成音

実験ごとの合成音を
集める



AI がテストを作る

系の名前を伏せ、順序を
入れ替えた MUSHRA、MOS、
AB 選好を自動で生成



Azure Static Web Apps

URL を配るだけで回答を集める



Blob Storage

回答を保存し、スクリプトで
集計する



公平さの検査もエージェントが提案した

無音の長さの違いで系を聞き分けられてしまう可能性に気づき、刺激の作り方を直した。系を聞き分ける手がかりが残っていないかを
数値で確認している

出典：講演者の実験記録（2026-09）

評価ページも、エージェントが自動で作った

聴き手は URL を開いて答えるだけ。回答は保存から集計まで自動で進む



AI が生成した評価ページ（回答の前に聴く環境を確認する画面）

1. **聴く環境を最初に確認**
   使う機器などの条件を、回答の前に尋ねる

2. **名前を伏せて聴き比べ**
   系の名前を伏せ、順序を入れ替えて提示する

3. **回答は自動で集計**
   Blob Storage に保存し、スクリプトで集計

人が決めたのは評価の方針だけ。ページの作成から回答の集計まで、人は手を動かしていない
聴く環境も回答と一緒に記録されるので、結果を後から解釈しやすい

出典：講演者の実験記録（2026-09）

1本の実験が1日前後で終わり、人は評価と解釈に時間を使えるようになった

1本の実験の、ある1日

朝	日中	夜
人：結果を聴いて 次の方針を決める	エージェント：実装、 評価の準備、レポート	学習 （夜間に実行）

毎回そろうもの

- README と実験レポート
- コマンドとログ
- うまくいかなかった試み
- 評価の道具
（スクリプト、Web アプリ）

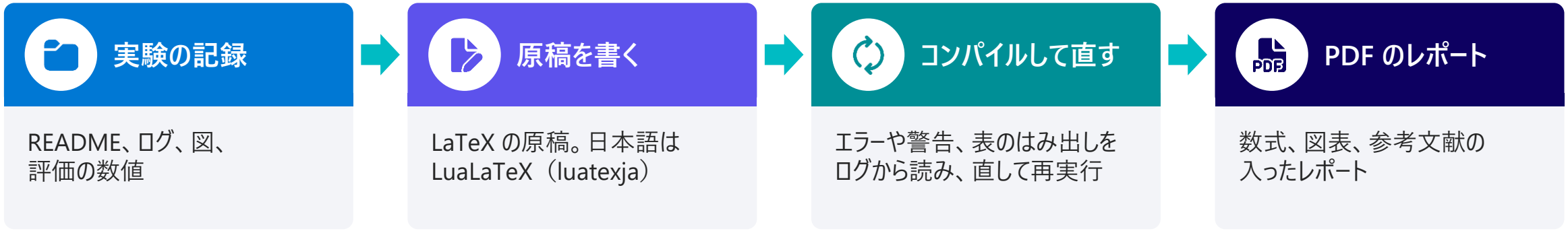
人が時間を使うようになったところ

- 何を良しとするかという評価の設計
- 数値の解釈と、次に何を試すかの選択
- データのライセンスと研究倫理
- 結論に対する責任

出典：講演者の実験記録（2026-09）

TeX の組版も、エージェントが引き受ける

実験の記録からレポートの原稿を書き、コンパイルのエラーも自分で読んで直す



6 本

音声合成の実験のレポート

実装 1 本、実験 5 本。実験を終えるたびに組版した

4 形式

アブストの下書き

同じ原稿から PowerPoint、Markdown、TeX（PDF）、Word を作る仕組みも、エージェントが作った

スキル

組版の手順を育てる

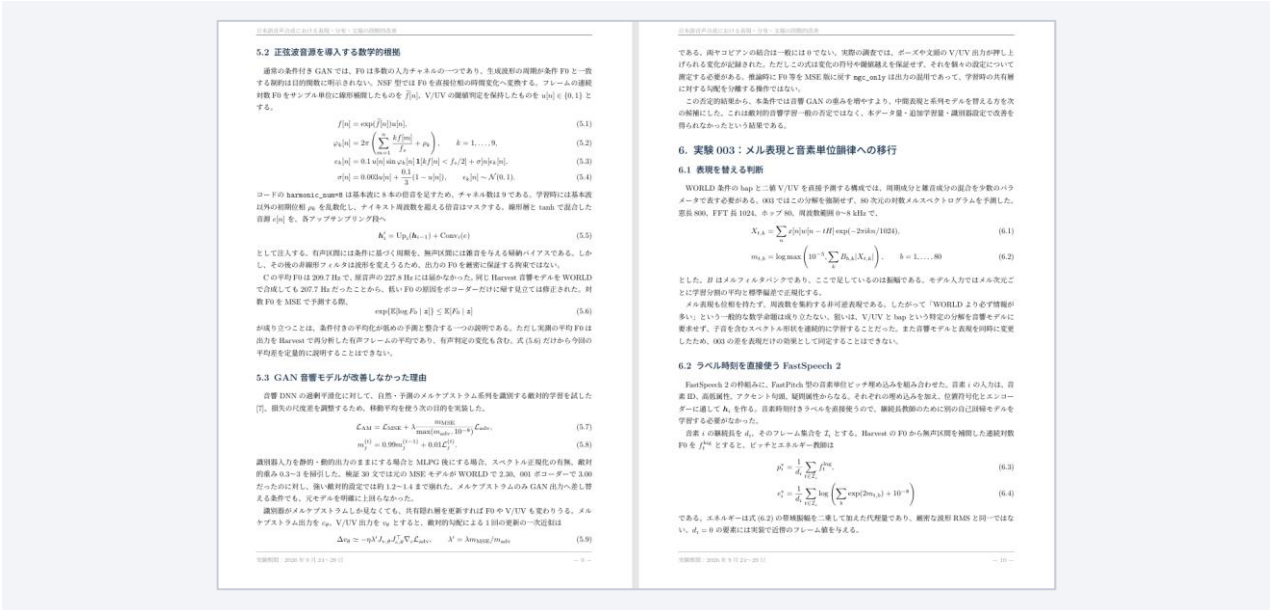
うまくいった手順と注意点をスキルに残し、次のレポートで使う

人が確かめるのは、数値が記録と合っているか、図の読み方、結論の書き方

出典：講演者の実験記録（2026-09）

エージェントが組版した実験レポートは、論文の体裁で仕上がる

実験の記録を渡すだけで、章立て、数式と式番号、考察までそろった PDF が出てくる



5.2 正弦波音源を導入する数学的根拠

通常の条件付き GAN では、F0 は多数の入力チャネルの一つであり、生成波形の周期が条件 F0 と一致する制約は目的関数に明示されない。NSF 型では F0 を直接位相の時間変化へ変換する。フレームの連続対数 F0 をサンプル単位に線形補間したものを $\tilde{f}[n]$、V/UV の閾値判定を保持したものを $u[n] \in \{0, 1\}$ とする。

$$f[n] = \exp(\tilde{f}[n])u[n], \tag{5.1}$$

$$\varphi_k[n] = 2\pi \left(\sum_{m=1}^{n} \frac{k f[m]}{f_s} + p_k \right), \qquad k = 1, \ldots, 9, \tag{5.2}$$

$$e_k[n] = 0.1\, u[n] \sin \varphi_k[n] \, \mathbf{1}[k f[n] < f_s/2] + \sigma[n] \varepsilon_k[n], \tag{5.3}$$

$$\sigma[n] = 0.003 u[n] + \frac{0.1}{3}(1 - u[n]), \qquad \varepsilon_k[n] \sim \mathcal{N}(0, 1). \tag{5.4}$$

コードの harmonic_num=8 は基本波に 8 本の倍音を足すため、チャネル数は 9 である。学習時には基本波以外の初期位相 p_k を乱数化し、ナイキスト周波数を超える倍音はマスクする。線形層と tanh で混合した音源 $e[n]$ を、各アップサンプリング段へ

$$\mathbf{h}'_i = \mathrm{Up}_i(\mathbf{h}_{i-1}) + \mathrm{Conv}_i(e) \tag{5.5}$$

として注入する。有声区間には条件に基づく周期を、無声区間には雑音を与える傾斜バイアスである。しかし、その後の非線形フィルタは波形を変えるため、出力の F0 を厳密に保証する再実ではない。

C の平均 F0 は 209.7 Hz で、原音声の 227.8 Hz には届かなかった。同じ Harvest 音響モデルを WORLD で合成しても 207.7 Hz だったことから、低い F0 の原因をボコーダーだけに帰す見立ては修正された。対数 F0 を MSE で予測する際、

$$\exp(\mathbb{E}[\log F_0 \mid \mathbf{a}]) \le \mathbb{E}[F_0 \mid \mathbf{a}] \tag{5.6}$$

が成り立つことは、条件付きの平均が弱めの予測と整合する一つの説明である。ただし実測の平均 F0 は出力を Harvest で再分析した有声フレームの平均であり、有声判定の変化も含む。式 (5.6) だけから今回の平均差を定量的に説明することはできない。

5.3 GAN 音響モデルが改善しなかった理由

音響 DNN の過剰平滑化に対して、自然・予測のメルケプストラム系列を識別する敵対的学習を試した[7]。損失の尺度差を調整するため、移動平均を使う次の目的を実装した。

$$\mathcal{L}_{\mathrm{AM}} = \mathcal{L}_{\mathrm{MSE}} + \lambda \frac{m_{\mathrm{MSE}}}{\max(m_{\mathrm{adv}}, 10^{-8})} \mathcal{L}_{\mathrm{adv}}, \tag{5.7}$$

$$m_i^{(t)} = 0.99 m_i^{(t-1)} + 0.01 \mathcal{L}_i^{(t)}. \tag{5.8}$$

識別器入力を静的・動的の出力のままにする場合と、MLPG 後にする場合、スペクトル正規化の有無、敵対的重み 0.3〜3 を掃引した。検証 30 文では元の MSE モデルが WORLD で 2.30、001 ボコーダーで 3.00 だったのに対し、強い敵対的設定では約 1.2〜1.4 まで悪化した。メルケプストラムの GAN 出力へ差し替える条件でも、元モデルを明確に上回らなかった。

識別器がメルケプストラムから見なくても、共有隠れ層を更新すれば F0 や V/UV も変わりうる。メルケプストラム出力を c_θ、V/UV 出力を v_θ とすると、敵対的勾配による 1 回の更新の一次近似は

$$\Delta v_\theta \approx -\eta \lambda' J_{v,\theta} J_{c,\theta}^{\top} \nabla_c \mathcal{L}_{\mathrm{adv}}, \qquad \lambda' = \lambda m_{\mathrm{MSE}} / m_{\mathrm{adv}} \tag{5.9}$$

である。両ケコピアンの結合は一般には 0 でない。実際の調査では、ポーズや文頭の V/UV 出力が押し上げられる変化が記録された。ただしこの式は変化の符号や優劣越えを保証せず、それを個々の設定について測定する必要がある。推論時に F0 等を MSE 版に戻す ngc_only は出力の混用であって、学習時の共有構に対する勾配を分離する操作ではない。

この否定的結果から、本条件では音響 GAN の歪みを増やすより、中間表現と系列モデルを替える方を次の候補にした。これは敵対的音響学習一般の否定ではなく、本データ量・追加学習量・識別器設定で改善を得られなかったという結果である。

6. 実験 003：メル表現と音素単位調律への移行

6.1 表現を替える判断

WORLD 条件の bap と二値 V/UV を直接予測する構成では、周期成分と雑音成分の混合を少数のパラメータで表す必要がある。003 ではこの分解を強制せず、80 次元の対数メルスペクトログラムを予測した。窓長 800、FFT 長 1024、ホップ 80、周波数範囲 0〜8 kHz で、

$$X_{t,k} = \sum_{n} x[n] w[n - tH] \exp(-2\pi i k n / 1024), \tag{6.1}$$

$$m_{t,b} = \log \max \left(10^{-8}, \sum_{k} B_{b,k} |X_{t,k}| \right), \qquad b = 1, \ldots, 80 \tag{6.2}$$

とした。B はメルフィルタバンクであり、ここで足しているのは振幅である。モデル入力ではメル次元ごとに学習分割の平均と標準偏差で正規化する。

メル表現も位相を持たず、周波数を集約する非可逆表現である。したがって「WORLD より必ず情報が多い」という一般的な数学命題は成り立たない。狙いは、V/UV と bap という特定の分解を音響モデルに要求せず、子音を含むスペクトル形状を連続的に学習することだった。また音響モデルと表現を同時に変更したため、003 の差を表現だけの効果として同定することはできない。

6.2 ラベル時刻を直接使う FastSpeech 2

FastSpeech 2 の枠組みに、FastPitch 型の音素単位ピッチ埋め込みを組み合わせた。音素 i の入力は、音素 ID、高低属性、アクセント句境、疑問属性からなる。それぞれの埋め込みを加え、位置符号化とエンコーダーに通して $\boldsymbol{h}_i$ を作る。音素時刻付きラベルを直接使うので、継続長教師のために別の自己回帰モデルを学習する必要がなかった。

音素 i の継続長を d_i、そのフレーム集合を $\mathcal{I}_i$ とする。Harvest の F0 から無声区間を補間した連続対数 F0 を $f_t^{\log}$ とすると、ピッチとエネルギー教師は

$$p_i^* = \frac{1}{d_i} \sum_{t \in \mathcal{I}_i} f_t^{\log}, \tag{6.3}$$

$$e_i^* = \frac{1}{d_i} \sum_{t \in \mathcal{I}_i} \log \left(\sum_{k} \exp(2 m_{t,k}) + 10^{-8} \right) \tag{6.4}$$

である。エネルギーは式 (6.2) の帯域振幅を二乗して加えた代理量であり、厳密な波形 RMS と同一ではない。$d_i = 0$ の要素には実装で近傍のフレーム値を与える。

実験期間：2026 年 9 月 21–29 日 — 9 —

実験期間：2026 年 9 月 21–29 日 — 10 —

エージェントが LuaLaTeX で組版した実験レポート（見開き 2 ページ）

1. **実験ごとに 1 章**
   「6. 実験 003」のように、実験の順に章が並ぶ

2. **数式と式番号まで組版**
   損失や特徴量の式を、番号付きで本文に入れる

3. **うまくいかなかった理由も残る**
   収束しなかった理由や、手法を替えた理由を書く

記録の読み込みから組版、エラーの修正まで、人は手を動かしていない
また、ここでエージェントと人にとって可読性の高いレポートを作ることが、次の実験でエージェントが読み返す「過去の知見」になる

出典：講演者の実験記録（2026-09）

研究室に持ち込むときの順番

最初にルールと過去の知見を並行してそろえ、残した記録を次の研究の知見に戻す

ルールを書く

AGENTS.md に、実行環境、記録の形式、データの扱いを書く

並行してそろえる

過去の知見を読める形に

論文、学位論文、実験ノート、失敗の記録まで、エージェントが検索して読める場所に集める

評価を先に固定する

ベースラインと評価スクリプトを先に作り、指標を落とさないことを約束にする

残すもの
評価スクリプトとベースラインの値

1 実験 1 ディレクトリ

連番のディレクトリに、目的、変更点、結果、考察を残す

残すもの
実験ごとのレポート

人は問いと判断に

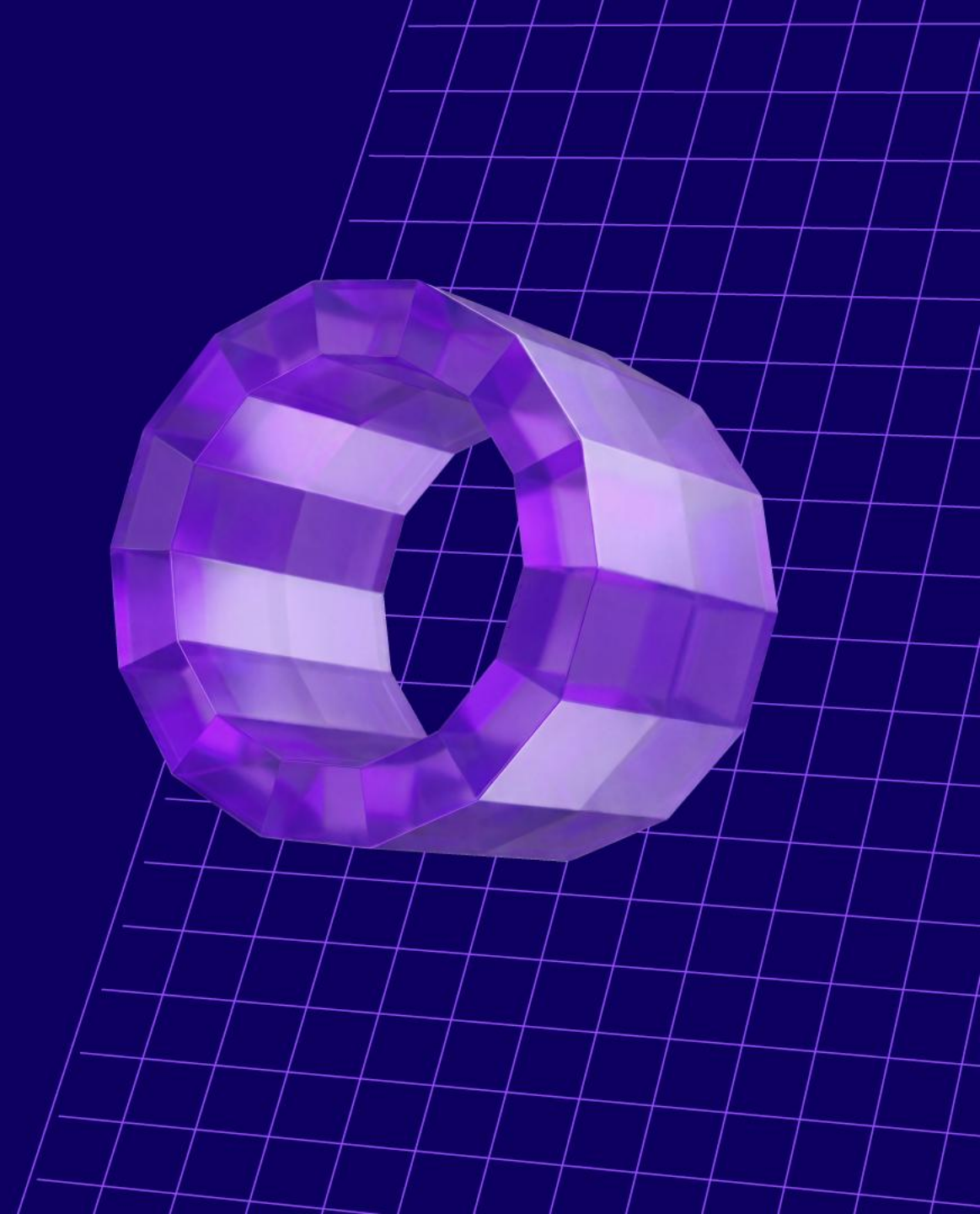
次に何を確かめるか、結果をどう解釈するかに時間を使う

残すもの
判断の理由と、次の問い

各段で残した記録は、次の研究で読む「過去の知見」に加わる

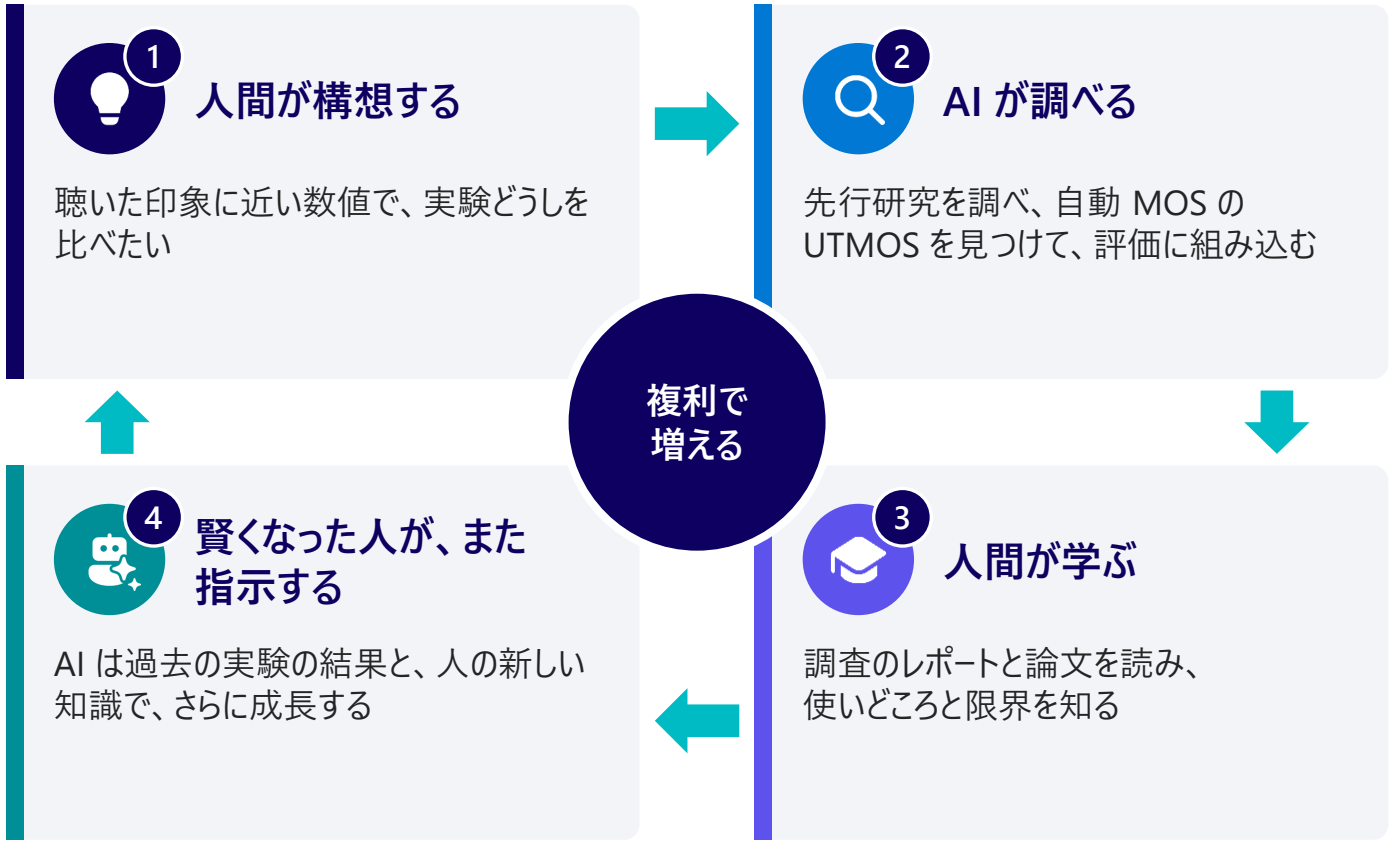
AI から学ぶ

人は AI から学べるか。任せたあと、人は何を担うか



AI が調べた結果から人が学び、人的資本とトークン資本が複利で増えていく

自動 MOS の UTMOS は、エージェントの調査で初めて知った



測る道具が、実験のたびに増えた

聴取テスト

+ MUSHRA、MOS、AB 選好。AI が作り、クラウドで配って集める

実験 000-004

+ UTMOS、MCD、F0 誤差、統計検定

最初の実装（9 月 25 日）

文字誤り率（CER）だけ。主観評価はなし

人が学び、AI が過去の結果と新しい知識で成長する。この複利のループが、人的資本とトークン資本の真骨頂

出典：講演者の実験記録（2026-09）、Saeki et al. 2022 (UTMOS)

試験対策では、AI の家庭教師が専門家の個別指導と同程度の効果を出した

査読前の論文

大学院進学の共通試験 GRE の学習で、2,383 人が 1 時間の指導を受けた実験

事前テスト
GRE 形式の 27 問

1 時間の指導
AI・専門家・指導なし

事後テスト
別の 27 問、AI なし

- 誤りを指導役に共有

- 3 群に無作為に割り当て
- AI は 13 種類の平均

- 得点の伸びで比較

+6.2 pt

AI の家庭教師 - 指導なし

27 問で約 1.5 から 2 問ぶん
95% 信頼区間は 4.1 から 8.2

+7.0 pt

専門家の個別指導 - 指導なし

27 問で約 2 問ぶん
95% 信頼区間は 3.9 から 10.2

-0.6 pt

AI と専門家の差

90% 信頼区間は -2.2 から +1.0
同等とみなす幅は ±4.1

AI と専門家の差は 0.6 ポイントで、同等とみなす幅の内側に収まりました。

出典：Northcutt et al. 2026 (StudentBench、査読前)

AI の時代に価値が高まるのは、人の判断

AI の便益を生かしながら、何を人が担うかを設計する。希少なのは実装の手数だけではない

AI に任せられる可能性が高まる領域

- 定型的な実装
- 情報の検索と整理
- 文書の初稿
- 定型的なテスト
- ドキュメントの作成
- 繰り返しの分析

時間と
注意を移す

人がより重要になる領域

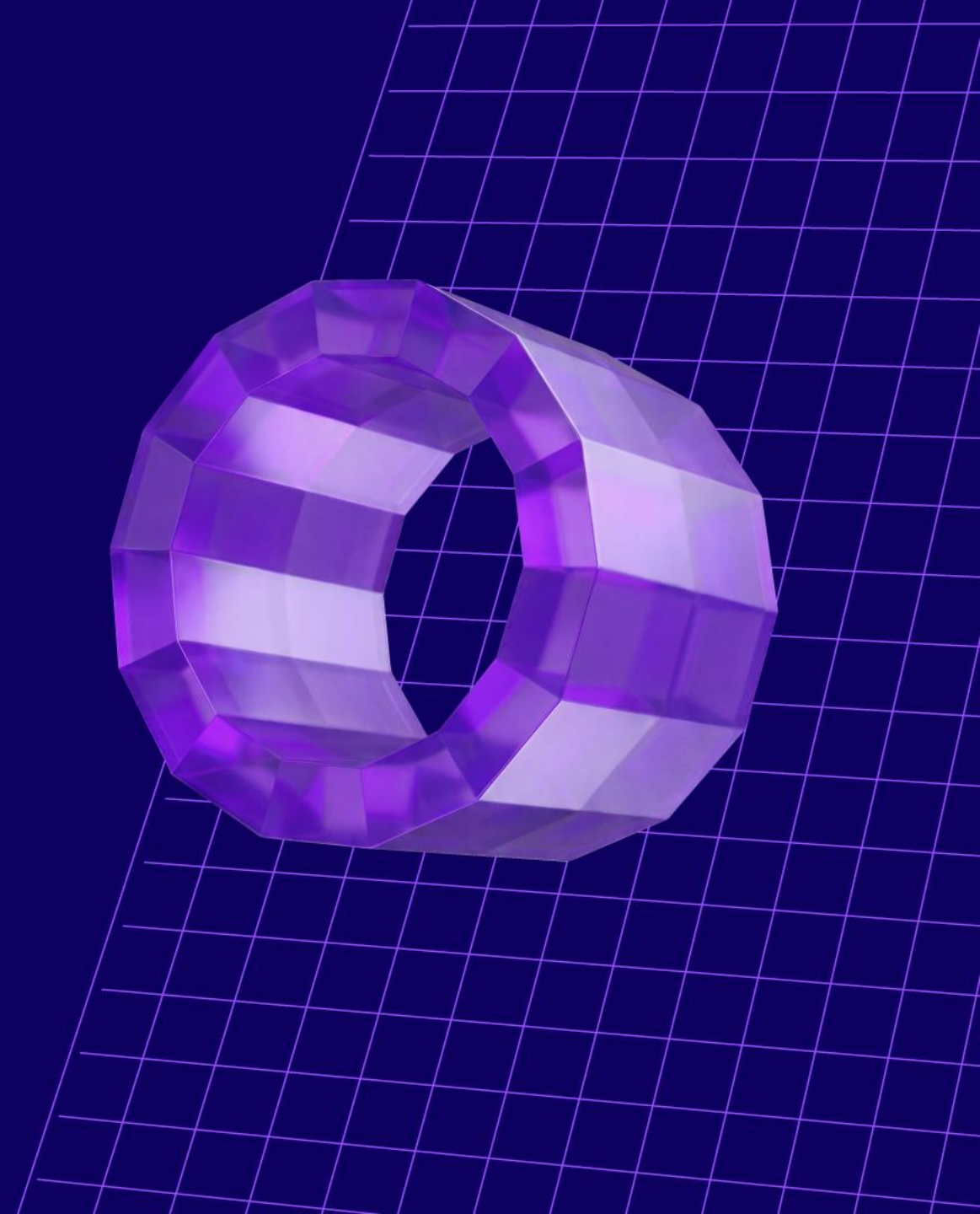
- 問題の定義と、科学的な判断
- アーキテクチャと評価
- 安全性と倫理
- ドメインの専門知識
- 何をつくるべきかの判断

作業を任せて生まれた時間を、人の判断、学習、関係づくりへ振り向ける

出典：Bill Gates / CNN SOTU (2026-10-11)

Microsoft Discovery

研究のサイクル全体をエージェントで回す



Microsoft Discovery は、研究開発のためのエージェント基盤

2025 年 5 月に発表、2026 年 6 月に一般提供。展開先のリージョンに Japan East を選べる

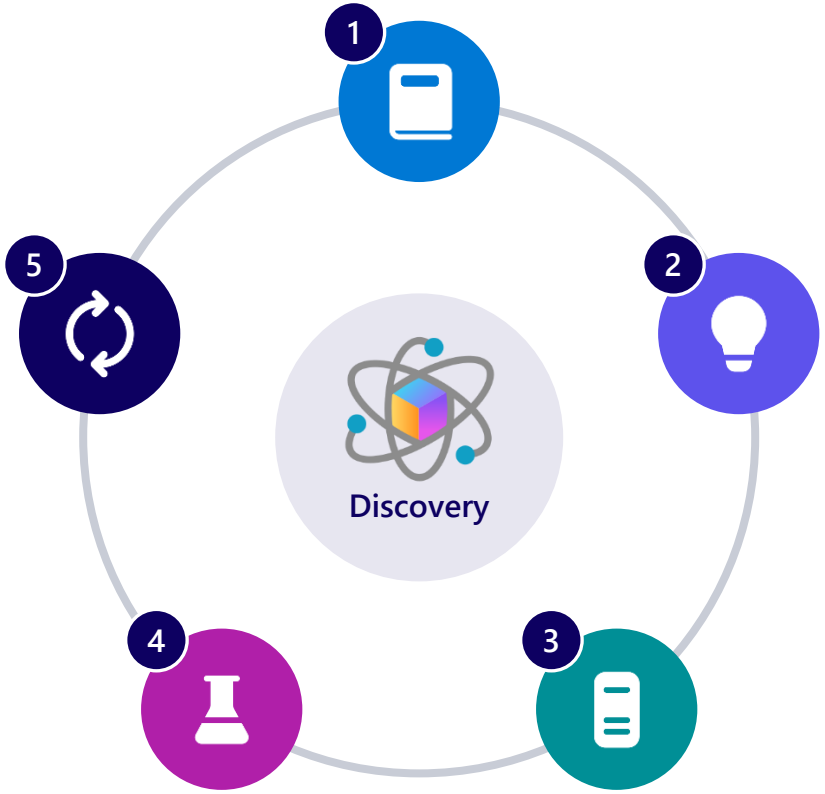


Microsoft Discovery

研究者の画面	Discovery Studio（Web）、API、CLI。目標を伝え、進み具合と根拠を確かめる	軽量な Discovery app はプレビュー
エージェントと推論	Discovery Engine が科学の手順をなぞり、仮説の生成と検証を繰り返す。専門のエージェントに作業を割り振る	Microsoft Research の研究（CLIO）が基盤
知識の基盤	Bookshelves（ナレッジグラフ）に文献、社内の知見、過去の結果を蓄え、グラフで推論する	結果を長期の記憶として蓄積
計算の基盤	AKS 上の Supercomputer が、コンテナにしたシミュレーションや解析の道具を実行する	Microsoft Foundry、Fabric と連携

出典：Microsoft Learn: Discovery の概要、Microsoft Learn: Discovery の構成、Azure Blog (2026-06, Discovery GA)、Microsoft Research Blog (2025-08, CLIO)

仮説、計算、実験、結果の還流までを一つのループで回す



1	**知識を集める**	文献、特許、社内の報告書、過去の実験をナレッジグラフに
2	**仮説を立てる**	候補の材料、条件、設計を、根拠とともに提案する
3	**計算で絞り込む**	シミュレーションや予測モデルで、多数の候補を評価する
4	**実験する**	有望な候補を合成・試作して測る。ロボットの自動実験とつなぐ例も
5	**結果を戻す**	結果を知識の基盤に戻し、次の仮説に使う

研究者は各段階で、方針の選択と結果の確認を担う。音声合成の実験の進め方を、組織の規模に広げたもの

出典：Azure Blog (2026-06, Discovery GA)

手元で育てたやり方を、組織の規模に広げる

個人

手元で探索する

Copilot CLI / Discovery App

- 知識の置き場はリポジトリ、AGENTS.md、スキル
- App はローカル・単一ユーザー向け

研究室

成果物とルールを共有

知識・手順・評価基準

- App の成果物を研究室で共有する
- 共同利用と権限、監査はクラウドで設計する

組織

研究開発全体の基盤

Microsoft Discovery

- 研究の知識をナレッジグラフで共有
- HPC と AI モデルで大規模に探索
- 権限の管理、監査、外部のラボとの連携

出典：Azure Blog (2026-06, Discovery GA)、Microsoft Learn: Discovery と Discovery app

公開されている事例では、数か月から数年の工程が、数日から数か月に縮んでいる

かかる期間の比較（横棒の長さはイメージ）

Microsoft と Submer

データセンター用の液浸冷却材（PFAS を使わない）

従来：数か月から数年

Discovery：約 200 時間　候補の発見まで。4 か月以内に試作品を合成

Microsoft Surface の設計チーム（PhysicsX）

冷却ファンの再設計

従来：数週間

Discovery：数日　シミュレーションの準備と実行。数千の候補を評価

BHP

銅の浸出技術の探索

従来：数年

Discovery：数か月

2026 年 6 月　Microsoft

量子チップ Majorana 2

製造の管理と、約 20 年分の実験データの分析に Discovery のエージェントを使った。信頼性を 1,000 倍に高めたチップの開発を支えたと説明している

自律実験室に近い例

- PNNL：ロボットと AI エージェントが、生物実験を設計・実行・調整する
- Ginkgo Bioworks：Discovery で計画した実験を自社のクラウドラボで実行する（予定）

出典：Azure Blog (2025-05, Discovery 発表)、Azure Blog (2026-04)、Azure Blog (2026-06, Discovery GA)、Microsoft (2026-06, Majorana 2 と Discovery)

ロボティクス研究での使いどころ

ロボットを実験の道具に使う

自律実験室とつなぐ

人の監督のもとで、ロボットや実験機器、IoT 機器を操作する（発展中のシナリオ）

結果をすぐ次の仮説へ

測定の結果を知識の基盤に戻し、次の実験を計画する

ロボットの材料や設計を探す

材料と部品を探す

グリッパの表面材料、センサーの素材、潤滑剤や冷却材。MatterGen（候補の生成）と MatterSim（原子レベルのシミュレーション）も使える

設計条件を絞り込む

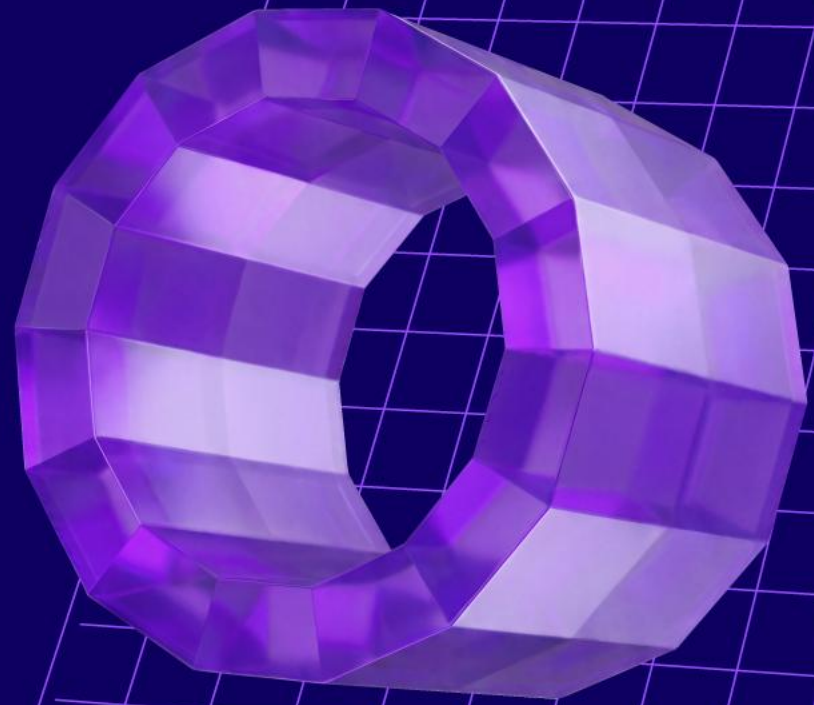
物理シミュレーションの条件や設計パラメータを、多数の候補から計算で絞り込む

ロボットは、実験を動かす道具にも、より良くする探索の対象にもなる

出典：Microsoft Learn: Discovery の主なシナリオ、Microsoft Foundry Labs: MatterGen、Microsoft Foundry Labs: MatterSim

Microsoft Discovery を活用する

Neural Subsumption × スキルマーケットの研究テーマを自動調査



調査テーマ：Neural Subsumption とスキルマーケットの統合

Discovery に自動調査させ、新規性・実現可能性・安全性・学術的価値を備えた研究テーマを設定

組み合わせる技術

Subsumption Architecture の行動階層と抑制に、
Neural Network による学習型スキルを組み合わせる

統合する仕組み

低価格な家庭用ロボットへ、第三者スキルを
追加できるスキルマーケット

人が与えたテーマ → **Discovery が調査** → **空白を特定** → **研究テーマを提案**

人が与えたテーマ	Discovery が調査	空白を特定	研究テーマを提案
研究目的と 4 つの評価観点を示す	先行研究を自動で収集・整理する	NN への単純な置換は新規性が不足	仮説・評価計画まで具体化する

新規性・実現可能性・安全性・学術的価値の 4 観点で、研究テーマを設定させた

出典：Discovery による自動調査（調査基準日 2026-07-16）、Microsoft Learn: Discovery と Discovery app

先行研究を領域ごとに整理し、研究上の空白を特定した

Neural Subsumption 単体は既知。複数領域の交点に空白がある

行動階層	スキル学習	安全保証	Edge 実装	スキル流通
Subsumption による行動の階層と抑制	Skill Library、MoE、異機体間の移植	Runtime Assurance、Simplex、CBF	TinyML と量子化で低価格機体へ載せる	Contract、署名、互換性判定、Rollback

初期調査の結論：組み合わせ自体は新規ではない

Togelius (2004) が層別進化による Neural Subsumption を提示済み。各行動層を NN に置き換えるだけでは新規性が不足する。

これらの交点、すなわち第三者 Neural Skill の安全な流通と実行に研究の空白がある

出典：Discovery による自動調査（調査基準日 2026-07-16）、Togelius, arXiv:cs/0405027

提案された研究テーマを確認し、必要なら人が修正する

提案の採否は人が持つ。新規性の根拠、実現可能性、安全性の主張範囲を確かめる

エージェントの提案を見る

- 研究テーマと中心仮説
- 5 層アーキテクチャと Skill Contract
- RQ、比較条件、評価指標の候補

人が確認・修正する

- 新規性の根拠となる先行研究は妥当か
- 300 ドル級の差動二輪で実現できるか
- 安全性の主張範囲と Go/No-Go 条件

モデルの提案は、前提・差分・検証で受け取る。実機の制御性能や機能安全は別途確かめる

出典：Discovery による自動調査（調査基準日 2026-07-16）、Nadella: Models as Insider Risks (2026-10-10)

見つけた研究テーマ：契約駆動 Neural Subsumption スキルマーケット

Contract-Driven Neural Subsumption Skill Marketplace for Low-Cost Household Robots

Skill Contract

能力・前提・効果・資源・安全包絡を宣言し互換性を判定

5 層アーキテクチャ

認証済みの L0/L1 が、未認証の Neural Skill を抑制する

異機体互換性

Adapter で Wheelbase や FOV の差を吸収して移植する

実行時安全保証

CBF/QP と Fallback で危険な Action を修復・停止する

上位層が能力を追加しても、下位の安全層は無効化できない設計を中心に据えた

出典：Discovery による自動調査（調査基準日 2026-07-16）、生成レポート・学術論文版

自動調査を、研究計画と共有できる成果物へつなぐ

今回の Discovery の調査では、一般向けレポートと学術論文版を生成し、12 か月の段階的研究計画まで示した

Discovery が調査

先行研究、研究ギャップ、仮説を整理する

成果物を生成

一般向けレポートと
学術論文版（参考文献 17 件）

人が確認して進める

前提を確かめ、M6・M10 で
Go/No-Go を判断する

12 か月の研究計画

Robot A/B と L0 Reflex から始め、Contract、RTA、
Adapter を段階的に検証する

主張の範囲

実証済み成果は主張しない position paper。
実証前にシステマティックレビューが必要

単一の高精度 Policy ではなく、Robot Ecosystem を成立させる System Research と位置付けた

出典：Discovery による自動調査（調査基準日 2026-07-16）、Microsoft Learn: Discovery と Discovery app

低価格家庭用ロボットのための契約駆動 Neural Subsumption スキル流通基盤

A Contract-Driven Neural Subsumption Infrastructure for Safe Skill Distribution on Low-Cost Household Robots

研究構想・構造化文献レビュー

調査基準日：2026 年 7 月 16 日

要旨—家庭用ロボットの普及には、購入後に機能を追加できるスキル流通基盤が有効と考えられる。しかし、第三者が提供する Neural Policy は、機体差、計算資源、安全性、更新時の回帰、複数 Policy 間の干渉という問題を伴う。本稿は、Subsumption Architecture、モジュール型 Robot Learning、Cross-Embodiment Transfer、Runtime Assurance、Edge AI に関する arXiv 文献 17 件を構造化して検討し、既存研究が個別に扱ってきた成果と、スキル市場として未統合の課題を整理する。その上で、Skill Contract、認証済み反射層、Action Repair、機体 Adapter、資源 Admission Control からなる Contract-Driven Neural Subsumption Skill Marketplace を提案する。本稿の中心的主張は、上位の未認証 Neural Skill を低位の検証可能な安全層が抑制する構造により、機能拡張性と実行時安全性を分離して評価できるという点にある。さらに、差動二輪の低価格移動ロボット 2 機種を対象とする研究質問、比較条件、故障注入、統計解析、および Go/No-Go 基準を提示する。本稿は実証結果を報告するものではなく、再現可能な実験へ接続するための研究設計を提示する position and research-agenda paper である。

キーワード— Subsumption Architecture; Robot Skill Marketplace; Runtime Assurance; Cross-Embodiment Transfer; TinyML; Safe Robot Learning

Abstract— A post-deployment skill marketplace could expand the utility of household robots, but third-party neural policies introduce embodiment mismatch, resource overruns, unsafe actions, update regressions, and policy interference. This paper conducts a structured thematic review of 17 arXiv studies spanning subsumption-based neurocontrollers, modular and lifelong robot learning, cross-embodiment transfer, runtime assurance, and edge deployment. We identify a systems-level gap: these strands have not been integrated into an end-to-end infrastructure for distributing untrusted skills to low-cost physical robots. We propose a contract-driven neural subsumption architecture composed of machine-readable skill contracts, certified reflexes, runtime action repair, embodiment adapters, and resource-aware admission control. We formulate research questions, falsifiable hypotheses, baselines, fault-injection scenarios, metrics, and statistical analyses for two low-cost differential-drive robots. This article is a research proposal and position paper; it does not claim empirical validation.

1. 序論

家庭用ロボットの価値を購入時の固定機能だけで決めるのではなく、後から追加可能な *skill* の集合として設計できれば、長期利用、第三者開発、ニッチ用途への適応が可能になる。一方、Robot Skill は一般の Mobile App と異なり、誤動作が物理空間へ作用する。特に Neural Network による Policy は、分布外入力、敵対的入力、推論遅延、説明困難性、および機体固有の状態・行動空間への依存を持つ。したがって、モデルを download して実行するだけの Marketplace は、安全性と互換性の両面で不十分である。

Subsumption Architecture は、複雑な知能を行動層へ分解し、低位の反射的行動が高位行動を抑制できる構造を導入した。その思想は高速、頑健、局所的な障害処理に適するが、古典的実装は手設計された行動と固定優先順位に依存する。Neural Network を Skill として扱う場合、この抑制関係を「未認証 Neural Skill の実行時統制」として再定義する。

本稿の目的は、次の 4 点である。

1. Subsumption、モジュール型 Robot Learning、Cross-Embodiment Transfer、Runtime Assurance、Edge AI を横断する 17 件の arXiv 文献を構造化する。
2. 既存研究が扱っていない、Skill Contract、Admission Control、Action Repair、Adapter、Safety Envelope を含む統合課題を定義する。
3. 低価格家庭用移動ロボットを対象とした反証可能な研究質問、仮説、比較実験、評価指標を設計する。
4. 本研究の主張範囲、倫理的課題、妥当性への脅威、および段階的な中止基準を明示する。

2. レビュー方法

2.1 探索範囲

2026 年 7 月 16 日を基準日とし、arXivの cs.RO、cs.AI、cs.LG、cs.CV、cs.SE、cs.PL、および eess.SY を中心に探索した。検索概念は、(i) subsumption / reactive / hierarchical policy、(ii) skill library / atomic skill / lifelong learning、(iii) cross-embodiment / latent alignment、(iv) runtime assurance / Simplex / CBF / action repair、(v) TinyML / quantization / resource-efficient robotics、(vi) 機体転移」「安全保証」「省資源実装」へ符号化し、研究対象、主要貢献、本構想に残る空白を比較した。

2.3 方法上の限界

Marketplaceや App Store という直接語で検索結果が少なくても、関連研究が Library、Repository、Cloud Robotics、Capability として存在する可能性がある。また、2026 年の文献には査読前 preprint が含まれる。本稿の数値引用は abstract に記載された結果に限定し、再現性や効果量の妥当性を保証しない。市場規模、利用者受容、価格弾力性は本レビューの対象外である。

3. 関連研究

3.1 Neural Subsumption と Policy 合成

Togelius は層別進化により Subsumption Neurocontroller を構築し、モノリシック進化に対する発展性を検討した [1]。この研究は直接的な先行例であるが、小規模 Simulation を対象とし、安全保証や第三者配布を扱わない。Urain *et al.* は Composable Energy Policies を提案し、複数の確率的 Policy の積を最適化することで競合する行動目標を調停した [2]。これは固定 Priority を超える連続的 Arbitration の候補である。

3.2 Skill Library と継続学習

LRLL は LLM、経験 Memory、Skill Abstractor を用いて再利用可能な Library を継続的に成長させる [3]。AtomicVLA は Skill-Guided Mixture-of-Experts を用い、Atomic Skill の追加と動的 Expert Routing を行う。abstract では実世界の長期タスクおよび継続学習で baseline に対し 18.3% および 21% の改善を報告している [7]。LiMoDE は凍結 Expert と新規 Expert を動的に合成する [8]。GOLLUM は Skill ごとの解釈可能な列を増殖させ、物理 Hexapod 上で複数の Locomotion Skill を継続獲得した [9]。これらは Skill の追加性を支持するが、異なる開発主体による Package の安全な流通は対象外である。

3.3 Cross-Embodiment Transfer

XSkill は非ラベルの人・ロボット動画から共通 Skill Prototype を発見し、Diffusion Policy へ接続する [4]。Wang *et al.* は source/target Robot の状態・行動を共通潜在空間へ写像し、新機体側の Encoder/Decoder 学習による Policy Transfer を示した [5]。これらはメーカー横断 Skill の基盤となり得るが、対応 Hardware、資源上限、安全包絡を含む install-time compatibility は未定義である。

3.4 Runtime Assurance

SOTER は未認証の高性能 Controller、認証済み安全 Controller、Safety Specification を Runtime Assurance Module として構成し、第三者 Component に対応がある DroneCase Study で安全性を評価した [10]。MORTAR は Verification と probabilistic enumeration から CBF を構成し、任意 RL Policy の危険 Action を補正した [13]。Lee *et al.* は高位 Multi-Agent RL と低位 QP を分離し、関節制約と衝突回避を低位で強制した [17]。

3.5 Contract Grounding と Edge 実装

Salfity *et al.* は Robot 側 MCP Server が Skill Library、許可 Behavior Tree Operator、Composition Template を Contract として提供し、Coding Agent が検証済み Behavior Tree を生成する構成を示した [14]。評価は 110 Simulation Task と 14 Physical Task を含む。TinyNav は ESP32 上で 23k parameter の量子化 CNN を用い、30 ms の推論時間を報告した [15]。Park *et al.* は 4-bit Quantization-Aware Imitation Learning により、実 Edge GPU の Robot Manipulation で最大 2.5 倍の高速化と省電力化を報告した [16]。これらは低価格実装の可能性を示すが、Skill Marketplace 全体の admission、更新、rollback は扱わない。

4. 問題設定

Robot を $r \in \mathcal{R}$、配布される Skill Package を $s \in \mathcal{S}$ とする。Skill Contract を

$$\mathcal{C}_s = \langle I, K, P, E, H, A, B, \Sigma, T, D \rangle \tag{1}$$

と定義する。ここで、I は Identity/Provenance、K は Capability、P と E は Precondition/Effect、H は Hardware Requirement、A は Action Interface、B は Resource Budget、Σ は Safety Envelope、T は Test Bundle、D は Telemetry/Data Policy である。

Robot Profile $\mathcal{P}_r$ と Contract $\mathcal{C}_s$ に対する install-time compatibility 判定を

$$\Gamma(\mathcal{P}_r, \mathcal{C}_s) \in \{\text{accept}, \text{adapt}, \text{reject}\} \tag{2}$$

とする。二値判定ではなく adapt を設ける理由は、Wheelbase、Sensor FOV、Action Range、Compute Budget の差を Adapter で吸収できる場合があるためである。

実行時には、Neural Skill が提案する Action a_t^N をそのまま適用せず、安全集合 $\mathcal{X}_{\text{safe}}$ を維持する Filter F を介して

$$a_t^* = F(a_t^N, x_t, \mathcal{C}_s), \qquad x_{t+1} \in \mathcal{X}_{\text{safe}} \tag{3}$$

を要求する。F は CBF/QP、Action Repair、または certified fallback で実装できる。本研究の目的は、Task Utility、Safety Violation、Latency、Energy、Adaptation Cost の多目的最適化として

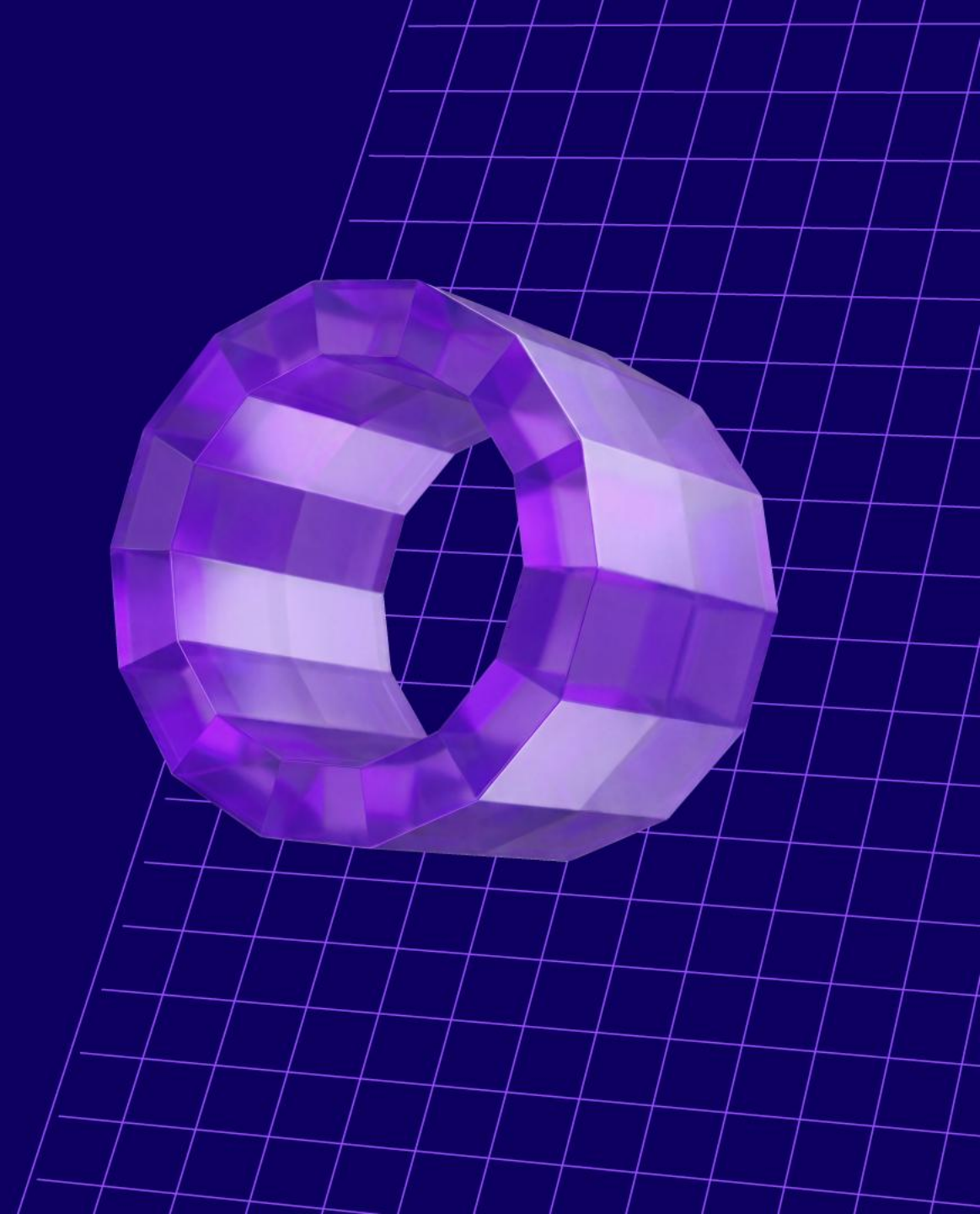
$$\max U_{\text{task}} - \lambda_1 V_{\text{safety}} - \lambda_2 L_{\text{ctrl}} - \lambda_3 E_{\text{step}} - \lambda_4 C_{\text{update}} \tag{4}$$

を評価することにあるが、ただし安全制約は単

Discovery が書き上げた学術論文版のレポート

データを見せずに
持ち寄る

現場のノウハウを守りながら、組織をまたいでデータを使えるか



現場のデータを持ち寄れないのには、技術の前に理由がある

個人情報

人が映る、声が入る

作業者の顔、声、行動が記録される。個人情報と労務管理への配慮が必要になる

営業秘密

工程とノウハウが見える

動作の軌跡や条件、図面、レシピから、競争力の源泉が推測できてしまう

権利

誰のデータか決まっていない

装置メーカー、ユーザー企業、SIer のどこに権利があるか、契約で決まっていないことが多い

形式

形式がそろっていない

時刻、座標系、単位、ラベルの付け方が現場ごとに異なり、そのままでは混ぜられない

出典：内閣官房「AI ロボティクス戦略」(2026-03)

国内の大規模なロボットデータでも、匿名化は前提の処理になっている

AI ロボット協会（AIRoA）が 2026 年 8 月に公開したデータセット AIRoA MoMa 5k

5,025 時間

ロボットの動作データ

成功したエピソード約 118 万件、約 4 TB。映像、ロボットの状態、力覚などを含む

44 台・5 拠点

収集の規模

トヨタの HSR を遠隔操作。2025 年 4 月から 2026 年 1 月に収集



組織をまたいで継続的に集めるなら、匿名化の処理を自動化し、記録を残す仕組みが要る

ぼかしを入れた映像

8.6%

約 237 万本の映像のうち、約 20 万本

人物検出で人が写っていると判定された映像にぼかしを入れた。映像は削除せず、ロボットの状態や動作のデータはそのまま残す

利用は研究開発目的に限り、再配布は禁止。データの公開と、プライバシーや権利の保護を両立させる設計

出典：AIRoA MoMa 5k (Hugging Face)、AIRoA (2026-08)

人が特定されるリスクは匿名化で下げられるが、会社のノウハウはデータの中身そのもの

匿名化・仮名化でリスクを下げられるもの

- 写り込んだ人の顔や姿（ぼかし）
- 声や話し方の特徴（音声の加工）
- 作業者や機器の ID（仮名化）
- 注意：体形、背景、時系列との突き合わせで再び特定されることがある

匿名化では守れないもの

- ロボットの動作の軌跡、速度、力のかけ方
- 加工の条件、レシピ、段取りの工夫
- 不良や故障の起き方と、その傾向
- 学習したモデルの重みそのもの

ノウハウを守るには、データを匿名化するだけでは不十分。データを相手に見せずに計算する仕組みが要る

出典：Microsoft Learn: Multi-party data

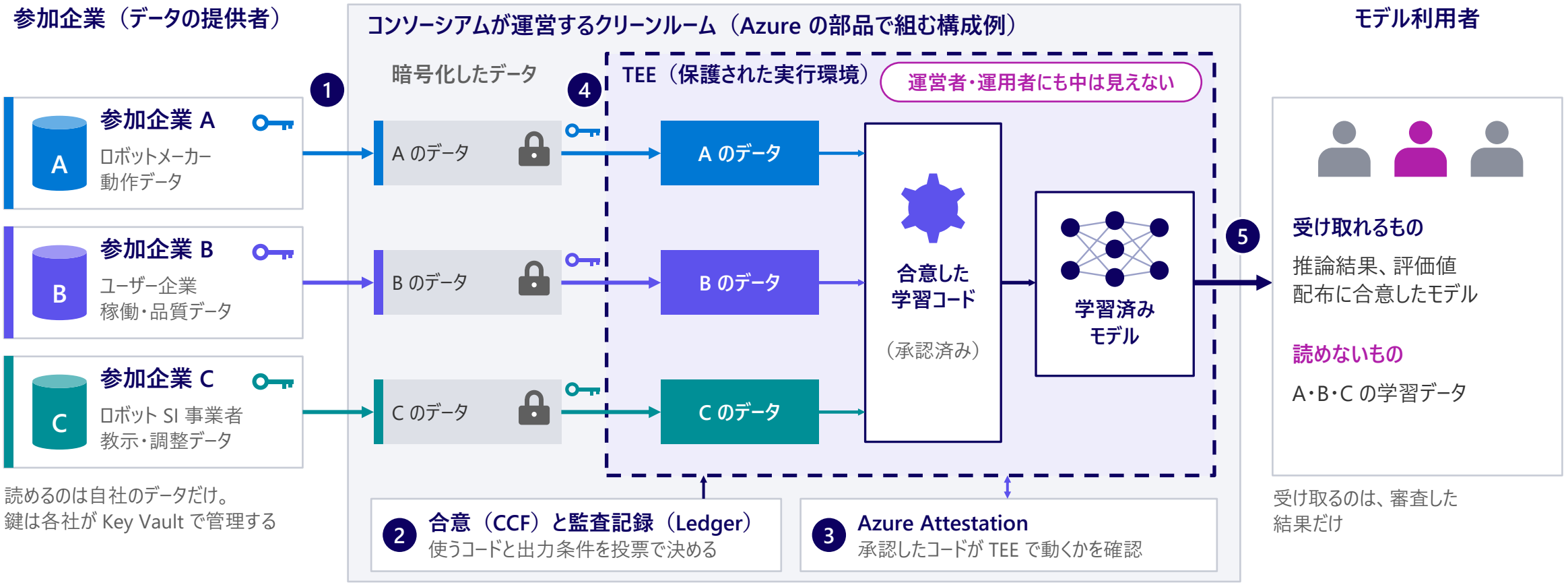
匿名化・秘匿化したまま研究データを整える方法は、目的と相手で使い分ける

	現場で匿名化して送る	連合学習	TEE を使うクリーンルーム	合成データ
共有するもの	顔や人物のぼかし、話者情報の除去、ID の仮名化を済ませたデータ	モデルの更新（重みや勾配）だけ。生データは各組織に残る	暗号化したデータ。合意したコードが TEE の中だけで復号して処理する	シミュレーションで生成したデータ。実データは検証に限って使う
守れるもの	**個人の特定リスク（法的な評価は別に必要）**	**生データを直接は渡さない（更新値からの漏えい対策は別に要る）**	**営業秘密、互いに見せたくないデータ**	**実データそのもの**
Azure での例	IoT Operations と Arc に処理を配備。Video Indexer の顔のぼかし、Presidio（OSS）	NVIDIA FLARE の集約役を Azure の機密コンテナで動かす手順（NVIDIA の文書）	機密 VM と機密 GPU、Azure Attestation、Key Vault など	Azure 上の NVIDIA Isaac Sim、Microsoft Foundry の Cosmos モデル

出典：Microsoft Learn: Video Indexer の顔のぼかし、Presidio (OSS)、NVIDIA FLARE ドキュメント、Microsoft Learn: NCC H100 v5、Microsoft Research (2026-01)、Microsoft Foundry: Cosmos3

データクリーンルームでは、データを見せずに持ち寄り、合意した計算だけを行う

データを復号できるのは、TEE の中で動く合意したコードだけ



出典：Microsoft Learn: Confidential Clean Rooms、Microsoft Learn: Multi-party data、Microsoft Learn: Azure Attestation、Microsoft Learn: Secure Key Release、Confidential Consortium Framework

誰が、誰のデータを読めるか

平文のデータに触れるのは TEE の中の合意したコードだけ。他社のデータを読める人はいない

読む人・読むもの	A のデータ	B のデータ	C のデータ	学習済みモデル（重み）	推論結果・評価値
参加企業 A ロボットメーカー	読める	読めない	読めない	合意次第	合意次第
参加企業 B ユーザー企業	読めない	読める	読めない	合意次第	合意次第
参加企業 C ロボット SI 事業者	読めない	読めない	読める	合意次第	合意次第
コンソーシアムの運営者 環境管理、記録保管、結果の審査	読めない	読めない	読めない	合意次第	合意次第
TEE の中の合意したコード 承認済みのプログラムだけが動く	読める	読める	読める	読める	読める
クラウドの運用者 Microsoft	読めない	読めない	読めない	読めない	読めない
推論サービスの利用者 入力を送り、結果を受け取る	読めない	読めない	読めない	読めない	読める
モデルの配布先 審査したモデルを受け取る	読めない	読めない	読めない	読める	読める

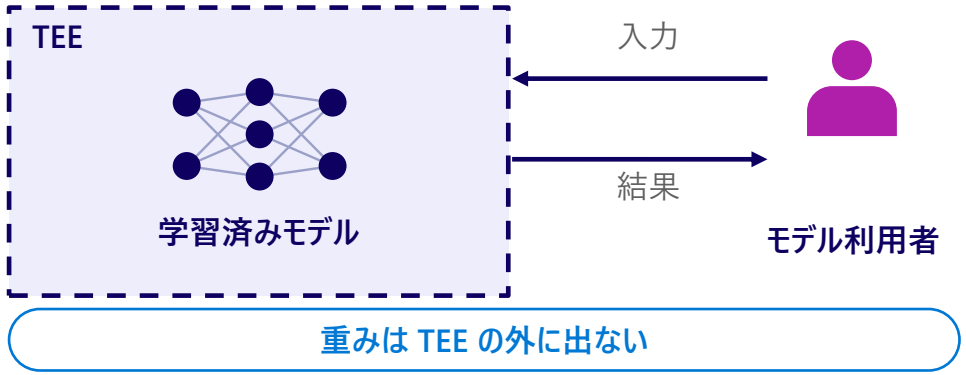
読める　平文で読める　　合意次第　参加者の合意で受け取れる　　読めない　読めない（暗号化されたまま、または渡されない）

出典：Microsoft Learn: Confidential Clean Rooms、Microsoft Learn: Multi-party data、Microsoft Learn: Secure Key Release

学習済みモデルの渡し方で、守り方が変わる

推論だけを提供すれば、重みは外に出ない。配布するなら、学習の段階から漏えい対策を組み込む

A　モデルは TEE に置き、推論だけを提供する

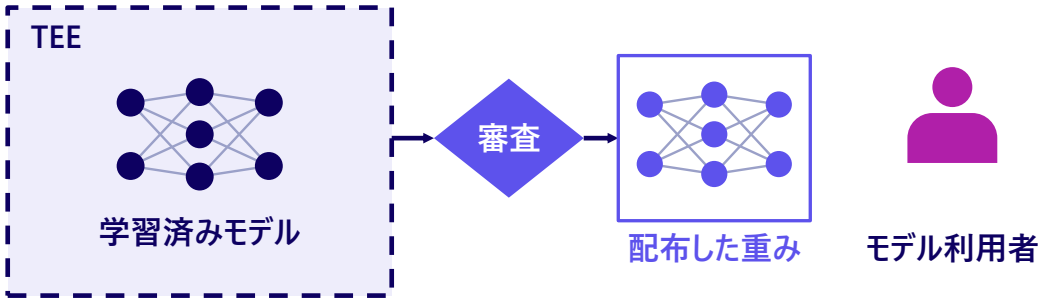


重みは TEE の外に出ない

利用者が見られるもの
自分の入力と、その結果だけ

注意点と対策
結果を大量に集めると、元のデータが推測できる場合がある。
問い合わせの回数や内容を制限し、結果を審査する

B　審査したモデルを配布する



重みが利用者の手元に渡る

利用者が見られるもの
モデルの重み（学習データそのものは含まない）

注意点と対策
モデルが学習データの一部を覚えていると、
重みから取り出せる場合がある。
差分プライバシーを使った学習、漏えいの試験、
配布先と用途の契約で備える

TEE が守るのは計算中のデータ。結果から何が分かるかは、出力の設計と審査で別に対処する

出典：Microsoft Learn: Multi-party data、Microsoft Learn: Confidential Clean Rooms

クリーンルームの部品は Azure にそろっている。まず集計と評価から試す

保護された実行環境（CPU）	機密 VM、ACI の機密コンテナ	提供中
保護された実行環境（GPU）	機密 GPU VM（NCCads H100 v5、H100 を 1 基）	提供中 ※
実行環境の証明	Microsoft Azure Attestation	提供中
鍵の管理と条件付きの受け渡し	Key Vault（Premium、Managed HSM）の Secure Key Release	提供中
参加者の合意と承認	Confidential Consortium Framework	OSS
改ざんを検知できる記録	Azure Confidential Ledger	提供中
マネージドのクリーンルーム	Azure Confidential Clean Rooms（複数組織の分析）	限定プレビュー

※ 2026 年 10 月 1-3 日時点の公開情報。提供地域は各製品ページを参照

いまの部品で試せること

- 複数社のデータを合わせた集計（最小件数などの条件付き）
- 非公開データでのモデルの共同評価（評価値だけを返す）
- GPU 1 基に収まるモデルの追加学習
- 連合学習の集約役を機密コンテナで動かす

制限と留意点

- 複数 GPU・ノードでの機密学習（対応状況は提供元の情報を参照）
- マネージド版の用途と制限（2026 年 10 月時点では分析の限定プレビュー）
- 結果からの漏えいを防ぐこと
- 参加条件、出力の審査、知的財産の取り決め

組み合わせ方、脅威の分析、セキュリティと法務の確認は利用者側で設計する。実データを預ける前に、合成データや公開データで試す

出典：Microsoft Learn: ACI の機密コンテナ、Microsoft Learn: Confidential GPU、Microsoft Learn: NCC H100 v5、Microsoft Learn: Azure Attestation、Microsoft Learn: Secure Key Release、Confidential Consortium Framework、Microsoft Learn: Confidential Ledger、Microsoft Learn: Confidential Clean Rooms、NVIDIA FLARE ドキュメント

匿名化は現場の近くで、統制は集約点で、学習はクラウドの機密コンピューティングで行う

1 現場



ロボット、PLC、カメラ、センサ。
データは各組織の拠点にある

2 エッジで前処理



Azure IoT Operations と
Azure Arc で収集し、匿名化と
形式の統一を済ませる

3 集約と統制



Microsoft Fabric に集約し、
Purview で利用範囲を管理する

4 学習と評価



Azure ML や AKS で
Isaac Sim／Lab と組み合わせて
学習する

5 評価したモデルを現場のエッジに戻し、運用のデータを次の学習に回す（Foundry Local など）

互いに見せないデータは別の経路でクリーンルームへ
TEE の中で処理を終え、承認した結果だけを外に出す

学習ループの参照実装
Physical AI Toolchain
（OSS）

組織をまたぐ連携は、Open Data Spaces（IPA、RRI、DSA、東京大学が 2025 年 10 月に発表）などデータスペースの仕様に合わせて設計する

出典：Azure IoT Operations (Learn)、Fabric Real-Time Intelligence (Learn)、Microsoft Purview (Learn)、Microsoft Learn: Confidential Clean Rooms、Physical AI Toolchain (GitHub)、DSA (2025-10, Open Data Spaces)

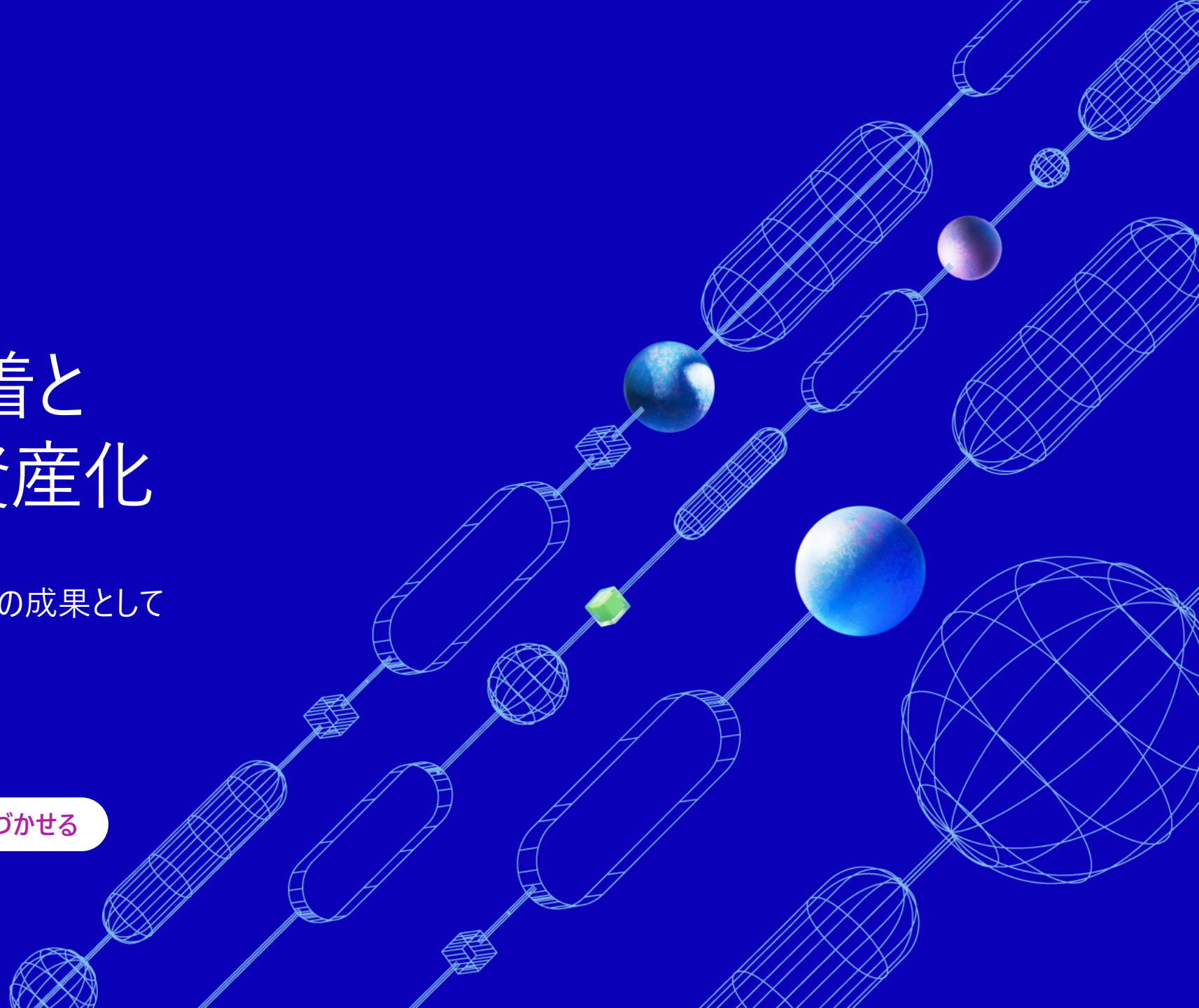
Microsoft Azure

AI の組織への定着と AI による知見の資産化

問い：個人の効果を、どうすれば組織の成果として定着させられるか

今日の 4 つのトピック

- 届ける
- 作る
- 確かめる
- 根づかせる



このトピックの流れ

問い：個人の効果を、どうすれば組織の成果として定着させられるか

個人の効果と組織の成果	問い：個人の効果は、そのまま組織の成果になるか
使い続ける条件	問い：何をそろえれば使われ続けるか。日本はどこにいるか
組織の知見を資産にする	問い：誰でもすぐ作れる時代に、組織に何を積み上げるか
組織文化と AI	問い：定着を左右するのは、人の意欲か、組織の仕組みか
定着と効果の測り方	問い：利用の広がりと、成果をどう測るか
エージェント時代の進め方	問い：エージェントの時代に、何から、どの順で進めるか

AI の効果は、課題と人で違い、実務に近づくほど小さく見える

実験と評価

現場実験と政府の評価で、これまでに確かめられていることです

実験室 得意な作業を選んだ実験	職場の試験 56 社と 66 社の RCT	政府の評価 英国政府の試行と評価
▪ 所要時間が 26 から 73% 短縮 ▪ 品質は多くの研究で差なし	▪ メールを読む時間が週 30 分減 ▪ 効果は一人で完結する作業に集中	▪ 時間の削減の多くは本人の申告 ▪ 英国 DBT は生産性向上の証拠なし

+14%

問い合わせの解決件数

初心者は +34%、熟練者はほぼ変化なし

25.1%

課題を終える時間の短縮

AI が不得意な課題では、正答が減少

+19%

熟練開発者の作業時間

本人の体感は 20% の短縮

個人で効果が出ても、それがそのまま組織の成果になるとは限りません。

出典：Microsoft Research 第1報 (2023-12)、Dillon ほか 2025 (Early Impacts of M365 Copilot)、Dillon ほか (NBER w33795)、英国 GDS 横断評価 (2025-06)、英国 DBT 評価 (2025-08)、Brynjolfsson et al. 2025 (Generative AI at Work)、Dell'Acqua et al. 2023 (Jagged Technological Frontier)、METR (2025-07)

一人で完結する作業は変わったが、調整が必要な業務は変わらなかった

実験と評価

66 社・7,137 人の RCT の業務分析と、英国政府の評価です

変わったのは、一人で完結する作業

約 2 時間
週あたりのメールの時間の減少

12%
文書の完成までの速さの向上

減少
時間外の労働

変わらなかったのは、調整が必要な業務

変化なし
会議の時間

検出できず
業務の量と構成の変化

証拠なし
組織の生産性の向上

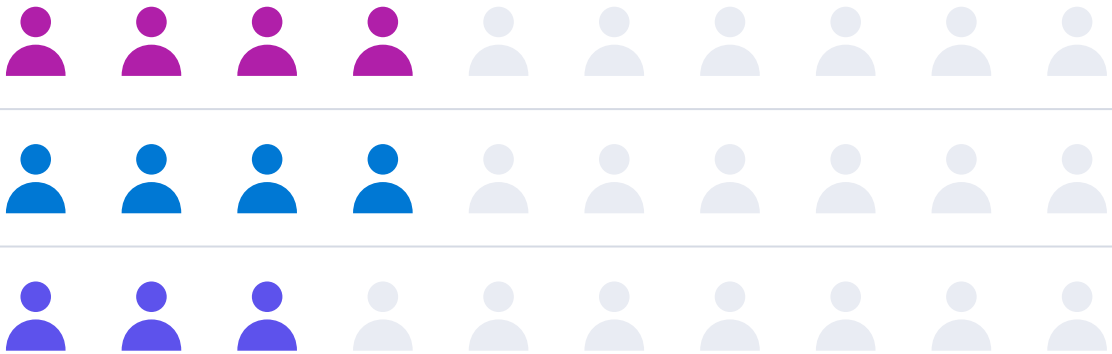
個人に配るだけではチームの働き方は変わりにくいため、チーム単位で業務の進め方を見直します。

出典：Dillon ほか 2025 (Early Impacts of M365 Copilot)、Dillon ほか (NBER w33795)、英国 DBT 評価 (2025-08)、Computing (2025-09, 英国 DBT)

ライセンスを配るだけでは、使われ続けない

実験と試行

最も危険な失敗は事故ではなく「無風」です。配布と数回の試用と費用だけが残ります



30-40%
一度も試さなかった開発者
Microsoft ほか 3 社の実験

約 40%
6 か月を通じて使い続けた人
6,000 人超の RCT

3 人に 1 人
毎日使った人
豪州政府の試行

60% 超
研修と後押しがあった職場の利用率
Accenture、1-2 か月後に横ばい

8.5% → 42.5%
リマインダー後の利用登録率
Microsoft の実験、各 2 週間

11 週
続けると習慣になる期間
1 日 11 分の削減で価値を実感

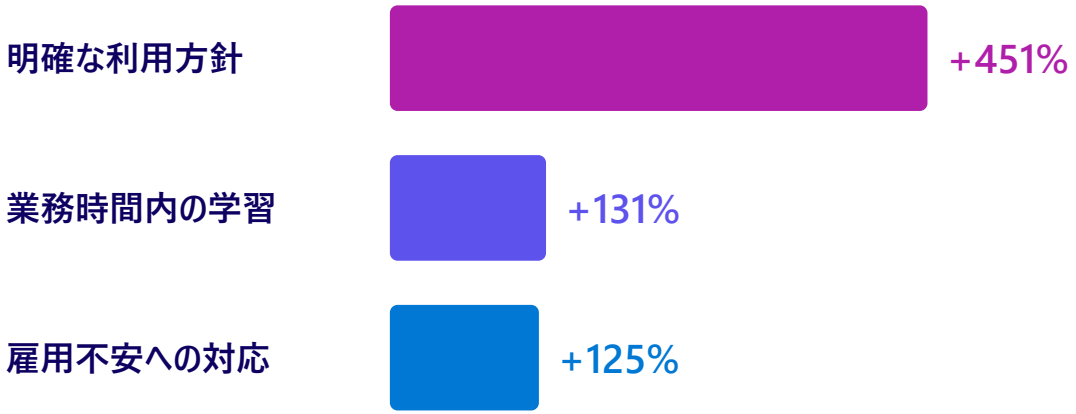
問うべきは「配ったか」ではなく「仕事が変わったか」です。

出典：Cui ほか 2025 (3 社の実験)、Dillon ほか (NBER w33795)、Dillon ほか 2025 (Early Impacts of M365 Copilot)、豪州 DTA 評価サマリー (2024)、GitHub・Accenture (2024-05)、Microsoft WorkLab 11×11、講演者の整理

定着と強く関連するのは、利用方針、業務時間内の学習、リーダーの関与

利用方針、学習の時間、リーダーの支援をそろえて定着を支える

DORA：チームの AI 採用（施策がない組織との比較）



業務時間内に学んで試す時間を確保する。私的な時間での学習を期待すると、不満や燃え尽きにつながる

15 → 55%
リーダーの強い支援
前向きな社員の割合（BCG）

67 → 79%
5 時間超の研修
定常的な利用（BCG）

+28 pt
3 種類以上の研修
使いこなす自信（豪州政府）

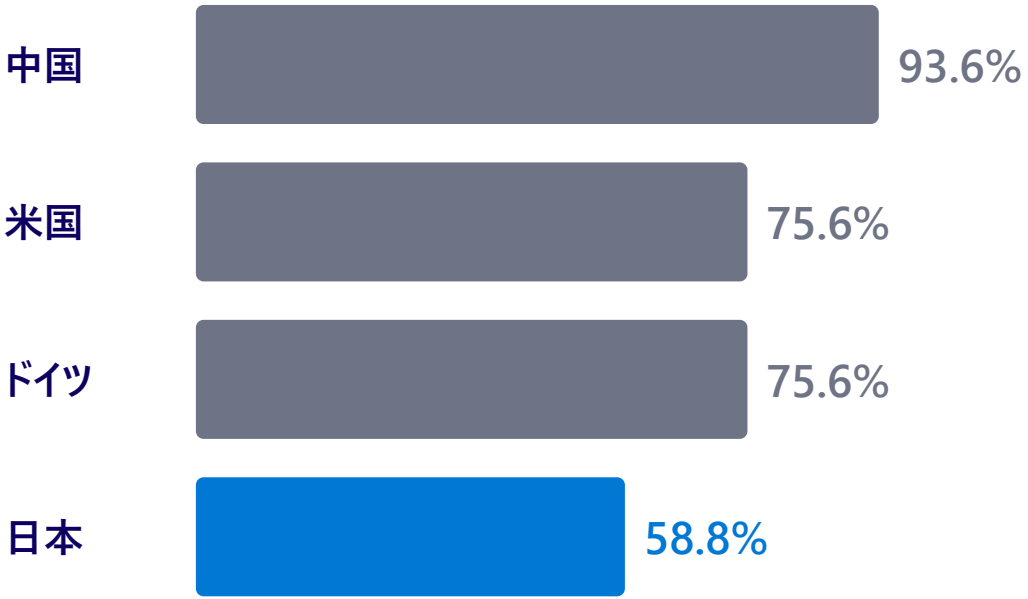
組織の条件と利用状況の関係を読み、自組織の施策は導入前後の変化で測る

出典：DORA: Impact of Gen AI、BCG AI at Work 2025、BCG AI at Work 2025 スライド、豪州 DTA (2024-10)、Cui ほか 2025 (3 社の実験)

日本の利用は急速に追いついたが、組織の取組と環境整備で差が残る

令和8年版 情報通信白書（2025 年度調査。日本・米国・ドイツ・中国）

生成 AI サービスを使ったことがある人（個人、2025 年度）



日本は前年度の 26.7% から上昇。15-19 歳を新たに調査対象に加えた（20 歳以上では 52.2%）

日本企業の状況

86.4% — **何らかの業務で生成 AI を使っている企業**
他の 3 か国との差も縮まった

68.9% — **生成 AI を活用する方針の企業**
前年度は 49.7%。それでも 3 か国より低い

27.0% — **組織的な取組がない企業**
米国・ドイツ・中国に比べて顕著に高い

日本の個人の利用経験率は、1 年で 26.7% から 58.8% に上がった。4 か国で最も大きな増加

出典：総務省 令和8年版 情報通信白書（個人の利用）、総務省 令和8年版 情報通信白書（企業の利用）

ゲームの構想を映像で公開した翌日に、AI で作った類似作が出た

報道と公開投稿

2026 年 10 月、開発中の映像が X で広まり、元の作品より先に Roblox で類似作が遊べる状態になりました

10 月 14 日

開発中の映像を X に投稿

壁を殴って脱出するゲーム
『Escape the Walls』

10 月 15 日

Roblox に類似作

よく似たゲームが複数公開。
制作者は AI に作らせたと発言

10 月 16 日

元の作品は発売前

元の投稿は約 450 万回表示。
Steam のストアページは準備中

8 月の例	「干し草の山から針を探す」ゲームの構想映像が広まり、48 時間で Roblox に模倣作が 14 本作られたと報じられました
国内の反応	10 月 4 日、国内のインディー開発者が、模倣を警戒して進捗の投稿を控えると表明しました

アイデアと見た目は、公開から数日で再現されうる時代になりました。

出典：Game*Spark (2026-10-16)、Pandorio Games (X, 2026-10-14)、Game*Spark (2026-08-25)、Game*Spark (2026-10-05)

研究の新しさも、公開すれば AI で追いつかれうる

論文

未公開の発見と同じ仮説を AI が挙げ、公開データからは定型の論文が急増しています

着想に追いつく
Cell, 2025

- 細菌の遺伝子伝達の仕組み
- 研究室が何年もかけて解明
- 未公開のまま AI に同じ問い
- 他の上位の仮説も、研究の糸口に

結果
AI の最上位の仮説が、実験の結論と一致

公開データから量産
PLOS Biology, 2025

- 米国の公開健康データ NHANES
- 単一の要因と病気の関連を見る論文
- 2014-2021 年は年平均 4 本
- 2024 年は 190 本（10/9 まで）

著者らの指摘
AI を使った量産と、誤った発見の増加

論文から再現
PaperBench, 2025

- ICML 2024 の論文 20 本が対象
- コードを書き、実験まで再現
- 最高は 21.0%（2025 年 4 月）
- 人の専門家を下回る水準

見方
論文を再現する力そのものを測定

研究は公開が前提です。公開したあとも先を走り続けられる、手元で積み上げる知見と仕組みが差になります。

出典：Penadés et al. 2025 (Cell)、Suchak et al. 2025 (PLOS Biology)、Starace et al. 2025 (PaperBench)、METR Time Horizon 1.1 (2026-01)

AI が手を動かす時代の強みは、組織に積み上がった知見の塊

講演者の整理

アイデアや公開した手順は追いつかれても、社内や研究室で積み上げた知見は真似しにくいものです

追いつかれやすいもの

- ⚠ アイデアと構想
- ⚠ 見た目と操作感
- ⚠ 論文に書いた手順
- ⚠ 公開したコードとデータ

積み上げた知見と、AI に渡す形

知見	真似しにくい理由	AI に渡す形
約束事	部署や研究室ごとの、見えない決まり	AGENTS.md、エージェントの指示
段取りと判断	手順書に残りにくい、ベテランの勘所	スキル
失敗の記録	公開されるのは、成功した結果だけ	実験記録、振り返り
評価の基準	論文に書き切れない、良し悪しの判断	評価のスクリプト、完了の条件
現場のデータ	取り方も含めて、社外に出さない資産	権限内で、エージェントの知識に

同じモデルでも、渡せる知見の厚みで結果が変わります。見せるのは成果、知見は内側で育てます。

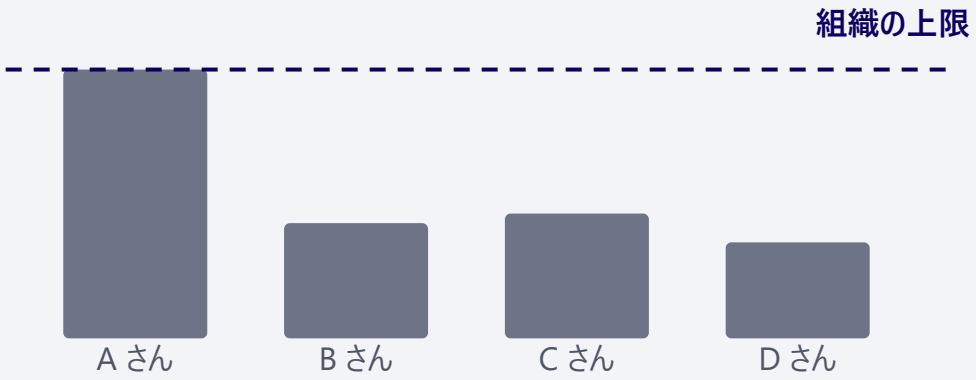
出典：講演者の整理

ベテランの暗黙知をスキルにして、組織のベースを上げる

個人の速さは、スキルにして初めて組織の力になる。手順書を人が読む形から、スキルを AI が実行する形へ

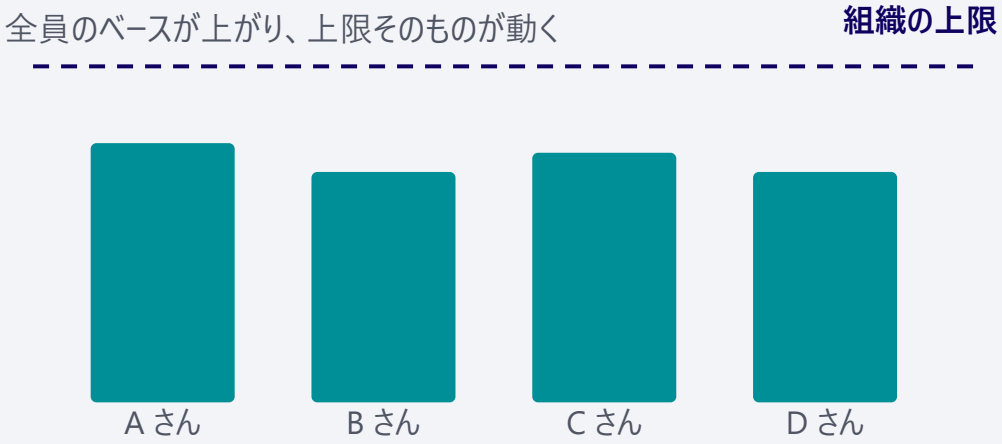
いま：属人化

一人の稼働が、組織の生産量を決めている



スキル化 + AI

全員のベースが上がり、上限そのものが動く



えらぶ

繰り返すが、手間のかかる業務を一つ

書き出す

ベテランの段取りを聞き、判断と観点を
言葉にする

スキルにする

登録して試し、使いながら直す

手順書を人が読む → スキルを AI が実行する。使うほど改善され、組織の資産として残る

出典：講演者の整理

Microsoft IQ と組織のスキルで、組織の文脈を AI に渡す

モデルは組織の経緯や関係者を知りません。足りないのは知識ではなく、仕事の周りの文脈です



組織のスキル	できる人の段取りと判断をスキルにして、AI が手順どおりに実行します
権限	Work IQ は既存の権限を保ったまま、Microsoft 365 のデータを扱います

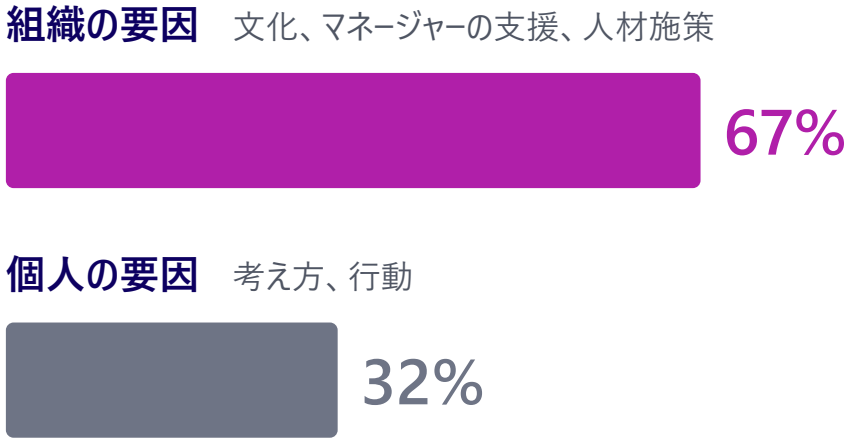
四つの IQ と組織のスキルで、文脈と仕事の進め方を共有の資産にします。

出典：Microsoft Learn: Fabric IQ の概要、Microsoft Learn: Work IQ の概要、講演者の整理

定着を左右するのは、個人の意欲より、組織文化とマネージャーの関わり

Work Trend Index 2026（10 か国、AI を使う 20,000 人の調査と、Microsoft 365 の利用データの分析）

AI の効果への寄与（Work Trend Index 2026）



変革のパラドックス（AI の利用者の回答）

65% 使いこなさないと取り残されることを恐れている

45% 仕事を作り替えるより、いまの目標に集中するほうが安全だと感じる

13% 結果が出なくても、仕事を作り替える挑戦が評価されると答えた

AI との対話とつなげると： 粗さがしで AI が萎縮したように、詰める文化は人の試行も減らす。試行を評価し、失敗を責めない文化が要る

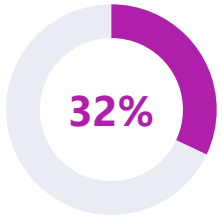
改善の工夫と試行を評価し、ノウハウが共有される文化をつくる

出典：Work Trend Index 2026

雇用の不安があると、AI は隠れて使われ、ノウハウが共有されにくい

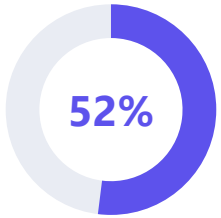
調査

上は Ivanti と KPMG の調査、下はそこから考えられる流れの整理です



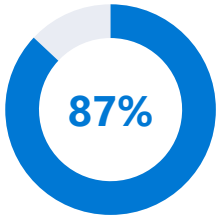
利用を雇用主に隠す

生成 AI を使う人のうち。Ivanti



仕事を置き換えられる懸念

前年からほぼ倍増。KPMG、米国



週 1 回以上 AI を利用

毎日使う人も 51%。KPMG、米国



仕事が減る不安

隠れて使う

共有されない

組織に残らない

使ってはいても、成果とノウハウは組織に共有されにくくなります。

出典：Ivanti (2025-05, Technology at Work)、KPMG (2025-11, American Worker Survey)、講演者の整理

効率化の成果が人減らしに使われると、人は改善に参加しなくなる

現場でよく聞く会話です（説明のためのイメージ）

職場の会話 説明のためのイメージ

マネージャー

どうして、業務の効率化に後ろ向きなの?

担当者

去年効率化したら、その分の人が減りましたよね。
忙しさは変わらず、給料も増えていません

マネージャー

.....

担当者

頑張って効率化しても、自分たちには
何も返ってこないんです

出典：講演者の整理（会話は説明のためのイメージ）

努力が報われなければ、改善は止まる

人は効率化にも AI にも後ろ向きになります。
では、どうすれば改善に参加するのでしょうか。

【過去の教訓】解雇しない協約が、同じ従業員を改善の担い手に変えた（NUMMI）

閉鎖された工場の同じ従業員が、「解雇しない」約束を境に改善へ参加するようになった

1982
前史：GM フリーモント工場が閉鎖。
欠勤率は約 20%、労使対立が深刻

1985
転換点：NUMMI 協約。旧工場の従業員を再雇用し、
存続を脅かす危機でない限り解雇しない。
解雇の前に役員・管理職の報酬を先に削る

1987-88
約束の試練：稼働率が 60% を下回っても、
誰も解雇せず、研修と改善活動に充てた

2008
波及：トヨタ北米の 2 工場で生産を 3 か月
止めても、待機の約 4,500 人を解雇せず、
研修と安全教育へ

NUMMI：協約の後の変化（Adler の研究）

仕事と環境に満足している従業員



欠勤率（低いほどよい）



改善提案制度への参加率



改善提案は 1 人あたり年約 8 件で、約 8 割を採用（ビッグ 3 の工場は 0.2 件、34%）

同じ人でも、職が守られれば改善に参加する。AI に対する取り組みでも同様の心理が働くであろうことは想像に難くない

出典：Adler 1993 (Time-and-Motion Regained, HBR)、Adler 1996 (Teams at NUMMI)、Workforce (2008-08-18, Toyota の待機 4,500 人)

「改善を理由に人を切らない」宣言、時間の再配置、評価の 3 本柱

AI で生まれた時間を、価値創造、育成、改善へ振り向ける

宣言

経営トップ名で「AI 活用による効率化を理由とした解雇は行わない」と発信。
適用範囲と例外（深刻な経営危機）を正直に明記する

時間の再配置

浮いた時間の行き先を、事前に決める：顧客対応の強化、新規事業・改善、リスキリング。
人員調整は自然減と配置転換で

評価・表彰

AI 活用事例の共有と提案件数を評価項目に。優れた事例を表彰して横展開する

改善の果実を人員削減に使えば、改善は止まる。人に還せば、改善は加速する。

測るのは目標値ではなく変化の向き：提案の件数と参加率、心理的安全性、時間の再配置先の内訳、AI 利用の申告率

約束するのは「絶対に切らない」ではなく「改善を理由に切らない」。守れる範囲で明確に

出典：講演者の提案（トヨタの教訓から）

定着と効果は、利用、行動、成果の順に測り、感覚だけで判断しない

利用 → 行動 → 成果 → 事業。体感と実測の両方で変化を捉える

利用

AI を使っている人の割合と、使う頻度

行動

どんな仕事を任せ、エージェントがどれだけ動いたか

成果

所要時間、品質、手戻りの量、出荷した成果物の数

事業

売上、コスト、顧客の満足、離職

感覚と実測は、食い違う

- 開発の実験では、体感と実測が逆になった（METR）
- 政府の試行では、時間削減の多くが本人の申告
- 利用が増えても、品質と安定は別に確かめる

Microsoft Copilot Dashboard の区分

- 準備：ライセンスと環境の整い具合
- 利用：アクティブな利用者と、機能ごとの使われ方
- 影響：仕事の仕方の変化。実感：サーベイ（Viva Glint など）の結果

利用の広がりに加えて、所要時間、品質、手戻り、事業への影響を測る

出典：Microsoft Learn: Microsoft Copilot Dashboard、METR (2025-07)、英国 GDS 横断評価 (2025-06)、DORA 2025

小さく測ってから広げる。効果を測っていない企業が多い

自己申告だけに頼らず、利用データと観察を組み合わせる

英国 DBT：4 つの方法を組み合わせた評価

- 利用データ
- 日誌調査
- 観察タスク
- 対照群へのインタビュー

結論：時間の削減が、生産性の向上につながった証拠は確認できなかった

LINEヤフー：測ってから全社に広げた

1. **試験導入**
   約 550 人・3 か月
2. **2 系統で測る**
   アンケートと GitHub の活動データ
3. **全社に展開**
   約 7,000 人

試験導入で、コーディング時間が 1 人 1 日あたり約 1-2 時間減

59.8%

言語系の生成 AI を導入した企業のうち、効果を測定していない企業
JUAS 企業 IT 動向調査 2025

指標は、個人の評価や報酬に結び付けない（GitHub ESSP の注意）

出典：英国 DBT 評価 (2025-08)、LINEヤフー (2023-10)、LINEヤフー Tech Blog (2025-04)、JUAS 企業IT動向調査2025

エージェントは、検証しやすい作業から試して広げる

講演者の整理

自動で検証しやすく、影響を限定できる作業から始めます

エージェントに任せる範囲 →

段階 1	段階 2	段階 3	段階 4
検証しやすい作業	**範囲の決まった変更**	**業務の流れの見直し**	**大きな変更と自律運用**
文書、テスト、小さな修正	機能の追加、定型のリファクタリング	会議、承認、知識の検索。チーム単位での進め方の作り直し	複数のエージェントの並行作業。評価、権限、監査、停止の仕組みの整備
自動で確かめやすい	レビューの容量が前提	チーム単位で価値	測り方と統制が課題

任せる範囲を広げるときは、レビュー、テスト、統制の体制も一緒に広げます。

出典：AIDev (Li ほか)、Anthropic (2026-01)、Faros AI (2025-07)、BCG AI at Work 2025、Dillon ほか (NBER w33795)

定着は、準備、有効化、測定と強化、流れの見直しの順に進める

講演者の整理

資料から組み立てた順序で、1 四半期を単位に繰り返して進めます

準備 配る前	有効化 最初の 11 週	測定と強化 1 四半期以降	流れの見直し その先
▪ データと権限の整理 ▪ 利用方針の決定 ▪ 使う作業と部門の選定 ▪ 導入前の状態の測定	▪ チーム単位での配布 ▪ 業務時間内に 5 時間以上の学習 ▪ 告知とリマインド ▪ 身近な相談役の配置	▪ 利用データと観察での測定 ▪ 使われない分の回収 ▪ レビューとテストの体制強化 ▪ 一時的な落ち込みを予算に反映	▪ チームの進め方の作り直し ▪ 検証しやすい作業からエージェントを試行 ▪ 育成の形の組み直し
変えたい仕事と比べる基準	チームでの使い方の習慣化	効果と詰まりの見える化	組織の仕事の流れの変化

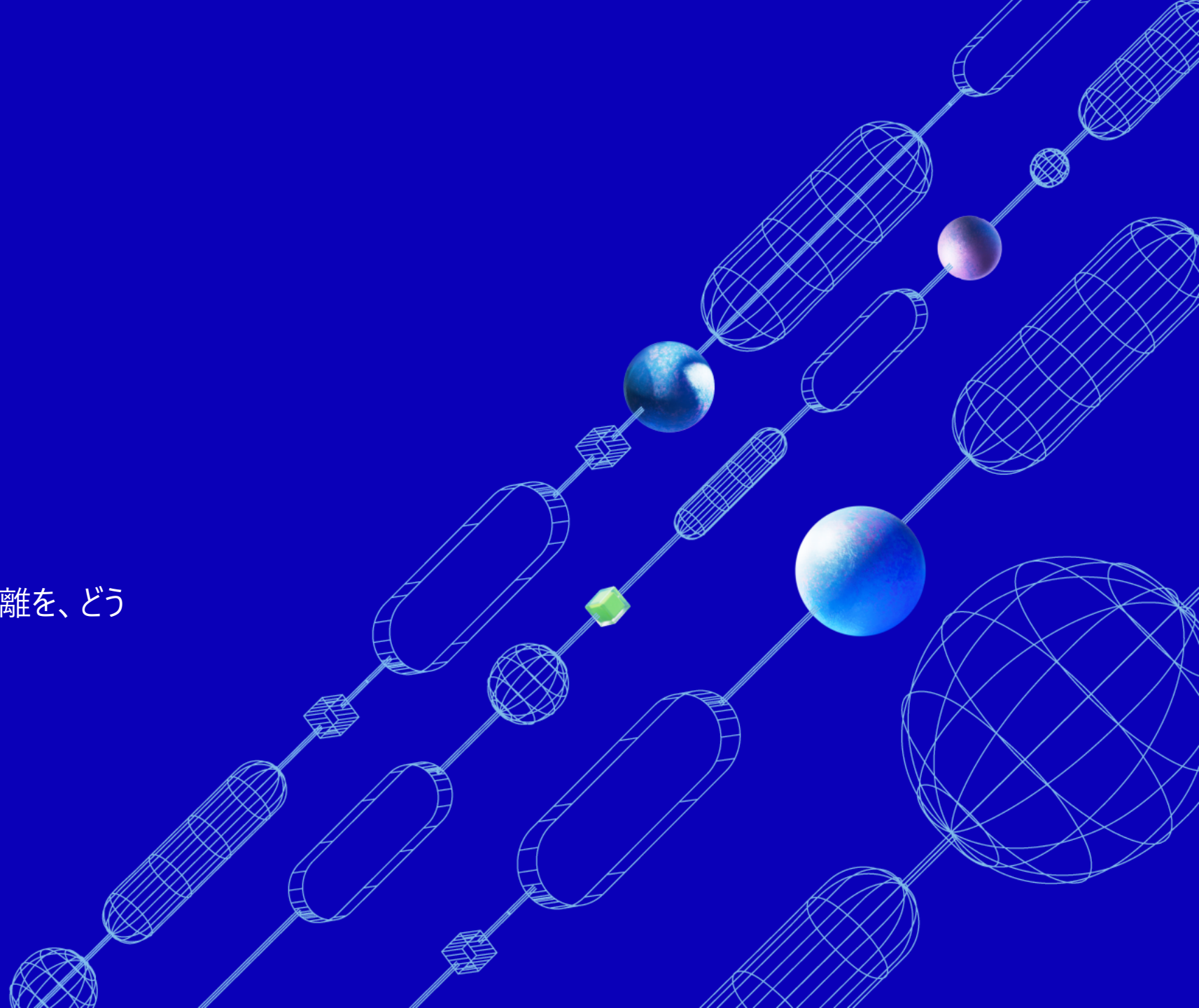
導入を一度きりで終わらせず、学習、測定、改善を繰り返します。

出典：Microsoft Learn: Copilot の安全な展開、DORA: Impact of Gen AI、JUAS 企業IT動向調査2025、Cui ほか 2025 (3 社の実験)、BCG AI at Work 2025 スライド、Microsoft WorkLab 11×11、Dillon ほか (NBER w33795)、BCG AI at Work 2025

Microsoft Azure

まとめと質疑

問い：アイディアが社会に届くまでの距離を、どう縮めていくか



まとめ：AI が、アイディアが社会に届くまでの距離を縮める

三つの場面で、任せられる範囲が広がり、人は問いと判断に時間を使えるようになってきた

業務

届ける

前提を伝えて任せ、記憶と指示を整える。AI は間違える前提で、事実確認と決裁は人が持つ

ソフトウェア開発

作る

書く作業から、仕様と確認へ。失敗を完成の定義に戻し、仕事の前提を指示とスキルで渡す

研究・開発

確かめる

評価を先に決めて実験を任せ、人は問いと判断を持つ。知識と手順は共有し、機密の現場データは見せずに持ち寄る

組織への定着

根づかせる

三つの場面を支える土台。積み上げた知見の塊を組織の資産にし、学ぶ環境と文化を整え、成果まで測る

任せる範囲を決め、確かめる仕組みを整え、問いと判断は人が持つ

明日から試せること

大きな投資をしなくても、いま使っている道具に、仕事の前提と知見を渡すところから始められます

Microsoft 365 Copilot
文書と調べものの仕事

- カスタム指示に、役割と書き方の好みを登録
- 保存されたメモリーを、ときどき見直し
- 資料を Notebooks に集め、質問して確認
- Agent Builder で、規程を読むエージェント
- 調べものは Researcher に依頼

最初の一歩
カスタム指示に、自分の役割を 3 行で書きます

GitHub Copilot と研究
コードと実験の仕事

- AGENTS.md に、実行環境と記録のルール
- 毎週の作業を一つ選び、スキルとして登録
- MCP で、文献の検索や実験データに接続
- レビュー役のカスタムエージェントを用意
- 評価を先に決めてから、実験を依頼

最初の一歩
研究のリポジトリに AGENTS.md を置きます

チームと組織
知見を積み上げる仕事

- 過去の論文や失敗の記録を、読める場所に
- うまくいった頼み方とスキルを、チームで共有
- 一つの業務で、導入の前後を測定
- 業務時間内に、学んで試す時間を確保
- うまくいかなかった例も共有し、試行を評価

最初の一歩
過去の報告書と実験記録を一か所に集めます

どれも、組織や研究室にしかない知見を、AI が使える形で積み上げる最初の一歩になります。

出典：Microsoft Learn: Copilot のパーソナル設定とメモリー、Microsoft Support: Copilot Notebooks、Microsoft Learn: 宣言型エージェントのベスト プラクティス、Microsoft 365 Blog (2025-06, Researcher)、GitHub Docs: カスタム指示、GitHub Changelog (2025-08, AGENTS.md)、GitHub Changelog (2025-12, Agent Skills)、DORA: Impact of Gen AI、講演者の整理

質疑・ディスカッション

ご質問に加えて、皆さんの現場の状況をお聞かせください



業務で、まずエージェントに任せてみたい作業は何か



エージェントに渡したい、仕事の前提（ルール、手順、道具）は何か



研究のサイクルのどこが詰まり、組織をまたぐデータをどう使うか



AI を組織に定着させるうえで、いま足りないものは何か

Microsoft Azure

生成 AI と共に
日本の未来を創りましょう！

